KMC11

CPU MICRO DIAGNOSTICS

MD-11-DZKCA-A

EP-DZKCA-A-DL-A

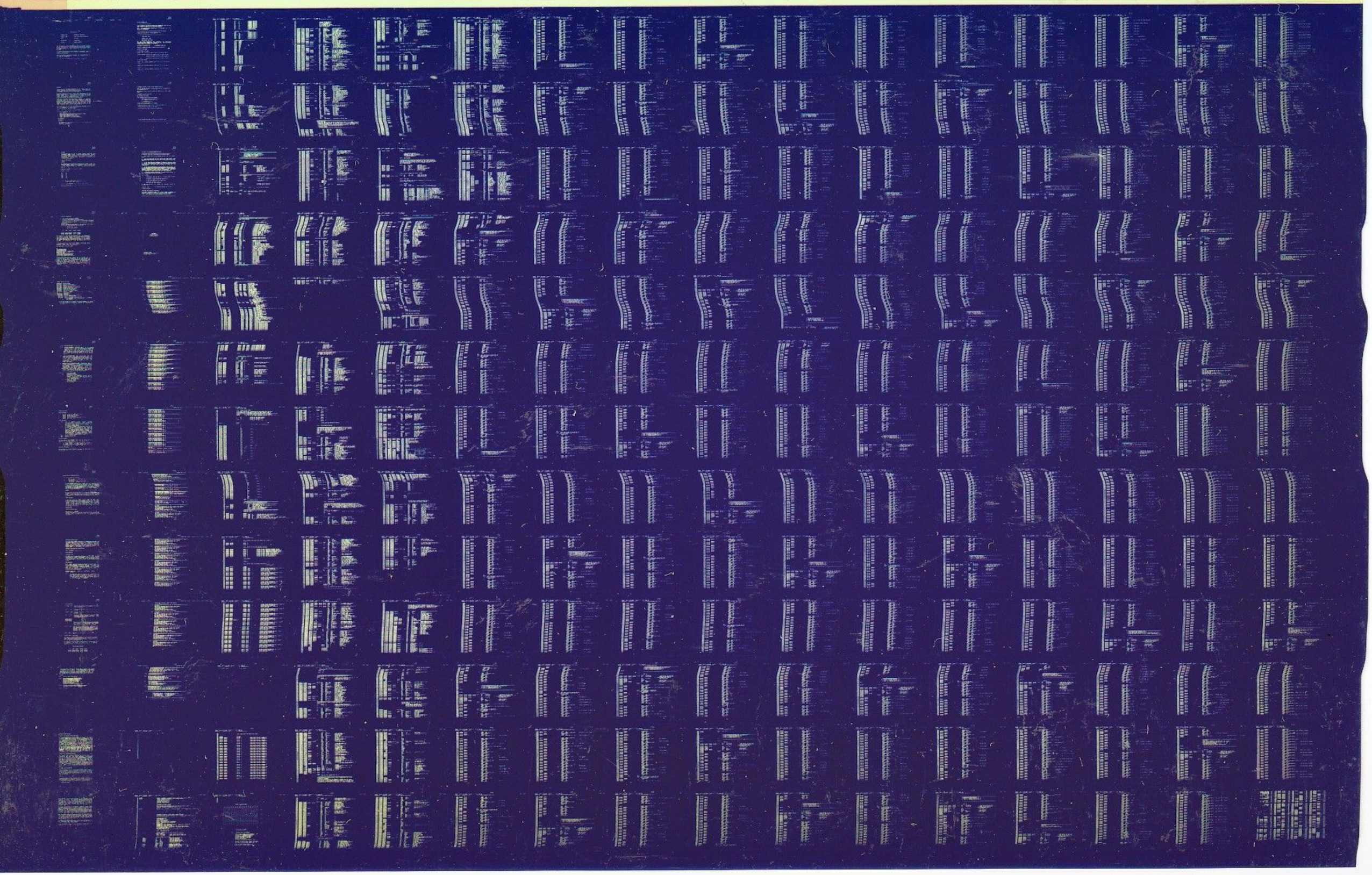
COPYRIGHT © 1977

FICHE 1 OF 2

AUG 1977

digital

MADE IN USA



KMC11

CPU MICRO DIAGNOSTICS

MD-11-DZKCA-A

EP-DZKCA-A-DL-A

COPYRIGHT © 1977

FICHE 2 OF 2

AUG 1977

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZKCA-A-D
PRODUCT NAME: KMC11 CPU MICRO-DIAGNOSTICS.
DATE: MAY 1977
MAINTAINER: DIAGNOSTICS
AUTHOR: DINESH GORADIA

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may only be used or copied in accordance with the terms of such license.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Digital.

Copyright (C) 1977 by Digital Equipment Corporation

1. ABSTRACT

The function of the KMC11 diagnostics is to verify that the option operates according to specifications. The diagnostics verfiy that there are no malfunctions and the all operations of the KMC11 are correct in its environment.

Parameters must be set up to alert the diagnostics to the KMC11 configuration. These parameters are contained in the STATUS TABLE and are generated in two ways: 1) Manual Input - the operator answers questions. 2) Autosizing - the program determines the parameters automatically.

DZKCA tests the KMC11 micro-processor (M8204). It performs write/read tests on the KMC unibus registers, checks the micro-processor operation, checks out Main Memory, scratch pad memory, the ALU functions as well as interrupts and NPR operation. DZKCA performs no tests on the line unit or any CRAM dependent tests. It will run on KMC11's containing CRAM (IOP). It does not require a line unit to run.

Currently there are four off line diagnostics that are to be run in sequence to insure that if an error should occur it will be detected at an early stage.

NOTE: Additional diagnostics may be added in the future.

The four diagnostics are:

1. DZKCC [REV] Basic H/R and Micro-processor tests
2. DZKCD [REV] Jump and memory tests (Heat test tape)
3. DZKCE [REV] DDCMP Line unit tests
4. DZKCF [REV BITSTUFF Line unit tests
5. DZKCA [REV] KMC11 CPU MICRO-DIAGNOSTICS.

2. REQUIREMENTS

2.1 EQUIPMENT

Any PDP11 family CPU (except an LSI-11) with minimum 8k memory
ASR 33 (or equilivalent)
KMC11-AN IOP (M8204)

2.2 STORAGE

Program will use all 8K of memory except where ABL and BOOTSTRAP LOADER reside. Locations 2100 thru 2300; contain the "STATUS TABLE" information which is generated at start of diagnostics by manual input (questions) or automatically (auto-sizing). This area is an overlay area and should not be altered by the operator.

3. LOADING PROCEEDURE

3.1 METHOD

All programs are in absolute format and are loaded using the ABSOLUTE LOADER. NOTE: if the diagnostics are on a media such as DISK ,MAGTAPE,DECTAPE, or CASSETTE; follow instructions for the monitor which has been provided on that specific media.

ABSOLUTE LOADER starting address #500

MEMORY	# SIZE
4k	17
8k	37
12k	57
16k	77
20k	117
24k	137
28k	157

3.1.1 Place address of ABS loader into switch register.
(also place 'HALT' SW up)

3.1.2 Depress 'LOAD ADDRESS' key on console and release.

3.1.3 Depress 'START KEY' on console and release (program should now be loading into CPU)

4. STARTING PROCEEDURE

a. Set switch register to 000200
b. Depress 'LOAD ADDRESS' key and release
c. Set SWR to zero for 'AUTO SIZING' or SWR bit0=1 for manual input (questions) or SWR bit7=1 to use existing parameters set up by a previous start or a previously run KMC11 diagnostic.
d. Depress 'START KEY' and release. The program will type Maindec Name and program name (if this was the first start up of the program) and also the following:

MAP OF KMC11 STATUS

PC	CSR	STAT1	STAT2	STA13
002100	160010	045310	177777	000000
002110	160020	045320	177777	000000

The program will type 'R' and proceed to run the diagnostic. The above is only an example. This would indicate the status table starting at add. 2100 in the program. In this example the table contains the information and status of two KMC11'S. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. For information of status table see section 8.4 for help.

If the diagnostic was started with SW00=1 indicating manual parameter input then the following shows an example of the questions asked and some example answers:

```
HOW MANY KMC11'S TO BE TESTED?1

01
CSR ADDRESS?160010
VECTOR ADDRESS?310
BR PRIORITY LEVEL?  (4,5,6,7)?5
WHICH LINE UNIT?  IF NONE TYPE "N",  IF  M8201  TYPE  "1",  IF
M8202 TYPE "2"?1
IS THE LOOP BACK CONNECTOR ON?Y
SWITCH PAC#1 (DDCMP LINE#)?377
SWITCH PAC#2 (BM873 BOOT ADD)?377
```

Following the questions the status map is printed out as described above, the information in the map reflects the answers to the questions. If the diagnostic was started with SW00=0 and SW07=0 (AUTO-SIZING) then no questions are asked and only the status-map is printed out. If AUTO-SIZING is used the status information must be verified to be correct (match the hardware). if it does not match the hardware the diagnostic must be restarted with SW00=1 and the questions answered.

4.1 CONTROL SWITCH SETTINGS

SW 15 Set: Halt on error
SW 14 Set: Loop on current test
SW 13 Set: Inhibit error print out
SW 12 Set: Inhibit type out abell on error.
SW 11 Set: Inhibit iterations. (quick pass)
SW 10 Set: Escape to next test on error
SW 09 Set: Loop with current data
SW 08 Set: Catch error and loop on it
SW 07 Set: Use previous status table.
SW 06 Set: Halt in ROMCLK routine before clocking micro-processor
SW 05 Set: Reserved
SW 04 Set: Reserved
SW 03 Set: Reselect KMC11's desired active
SW 02 Set: Lock on selected test
SW 01 Set: Restart program at selected test
SW 00 Set: Build new status table from questions. (If SW07=0 and SW00=0 a new status table is built by auto-sizing)

Switch 06 and 08-15 are dynamic and can be changed as needed while the diagnostic is running. Switches 00-03 and switch 07 are static, and are used only on starting or restarting the diagnostic.

4.1.2 SWITCH REGISTER OPTIONS (at start up)

SW 01 RESTART PROGRAM AT SELECTED TEST. It is strongly suggested that at least one pass has been made before trying to select a test, the reason being is that the program has to clear areas and set up parameters. When this switch is used the diagnostic will ask TEST NO.? Answer by typing the number of the test desired and carrige return to begin execution at the selected test.

SW 02 LOCK ON SELECTED TEST. This switch when used with SW01 will cause the program to constantly loop on the selected test. Hitting any key on the console will let it advance to the next test and loop until a key is hit again. If SW02=0 when SW01 is used. The program will begin at the selected test and continue normal operations.

SW 03 RESELECT KMC11'S DESIRED ACTIVE. Please note that a message is typed out for setting the switch register equal to KMC11's active. this means if the system has four KMC11s; bits 00,01,02,03 will be set in loc 'KMACTV' from the switch register. Using this switch(SW00) alters that location;therefore if four KMC11s are in the system ***DO NOT*** set switchs greater than SW 03 in the up position. this would be a fatal error. do not select more active KMC11s than there is information on in the status table.

METHOD: A: Load address 200
B: Start with SW 00=1
C: Program will type message
D: Set a switch for each KMC desired active.
EXAMPLE: If you have 4 KMC's but only want to run the first and the last set SWR bits 0 and 3 = 1. PRESS CONTINUE
E: Number (IF VALID) will be in data lights (excluding 11/05)
F: Set with any other switch settings desired.
PRESS CONTINUE.

4.1.3 DYNAMIC SWITCHES

ERROR SWITCHES

1. SW 12 Delete print out/bell on error.
2. SW 13 Delete error printout.
3. SW 15 Halt on the error.
4. SW 08 Goto beginning of the test(on error).
5. SW 10 Goto next test(on error).

SCOPE SWITCHES

1. SW06 Halt in ROMCLK routine before clocking micro-processor instruction. This allows the operator to scope a micro-processor instruction in the static state before it is clocked. Hit continue to resume running.
2. SW09 (if enabled by 'SCOPI') on an error; If an '*' is printed in front of the test no. (ex. #TEST NO. 10) SW09 is incorporated in that test and therefore SW09 is usually the best switch for the scope loop (SW14=0, SW10=0, SW09=1, SW08=0). If SW09 is not enabeled; and there is a HARD error (constant); SW08 is best. (SW14=1,0, SW10=0, SW09=0, SW08=1). for intermittemt errors; SW14=1 will loop on test reguardless of error or not error. (SW14=1, SW10=0, SW09=0, SW08=1,0)
3. SW11 Inhibit interations.
4. SW14 Loop on current test.

4.2 STARTING ADDRESS

Starting address is at 000200 there are no other starting addresses for the KMC11 diagnostics. (See Section 4.0)

NOTE: If address 000042 is non-zero the program assumes it is under ACT11 or XXDP control and will act accordingly after all available KMC11's are tested the program will return to 'XXDP' or 'ACT-11'.

5. OPERATING PROCEDURE

When program is initially started messages as described in section 4.0 will be printed, and program will begin running the diagnostic

5.2 PROGRAM AND/OR OPERATOR ACTION

The typical approach should be

1. Halt on error (via SW 15=1) when ever an error occurs.
2. Clear SW 15.
3. Set SW 14: (loop on this test)
4. Set SW 13: (inhibit error print out)

The TEST NUMBER and PC will be typed out and possibily an error message (this depends on the test) to give the operator an idea as to the source of the problem. If it is necessary to know more information concerning the error report; LOOK IN THE LISTING for that TEST NUMBER which was typed out and then NOTE THE PC of the ERROR REPORT this way the EXACT FUNCTION of the test CAN BE DETERMINED.

6. ERRORS

As described previously there will always be a TEST NUMBER and PC typed out at the time of an error (providing SW 13=0 and SW 12=0). in most cases additional information will be supplied in the the error message to give the operator an indication of the error.

6.2 ERROR RECOVERY

If for some reason the KMC11 should 'HANG THE BUS' (gain control of bus so that console manual functions are inhibited) an init or power down/up is necessary for operator to regain control of cpu. If this should happen; look in location 'STSTNM' (address 1202) for the number of the test that was running at the time of the catastrophic error. In this way the operator will have an idea as to what the KMC11 was doing at the time of the error.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

See section 4. (PLEASE)
Status table should be verified reguardless of how program was started. Also it is important to use this listing along with the information printed on the TTY to completly isolate problems.

7.2 OPERATING RESTRICTIONS

The first time a KMC11 diagnostic is loaded into core and run the STATUS TABLE must be set up. This is done by manual input (SW00=1) or by autosizing (SW00=0 and SW07=0). Thereafter however the status table need not be setup by subsequent restarts or even loading the next KMC diagnostic because the STATUS TABLE is overlayed. The current parameters in the STATUS TABLE are used when SW07=1 on start up.

7.3 HARDWARE CONFIGURATION RESTRICTIONS

KMC11 IOP(M8204)- Jumper W1 must be in,

8. MISCELLANEOUS

8.1 EXECUTION TIME

All KMC11 device diagnostics will give an 'END PASS' message (providing no errors and sw12=0) within 4 mins. This is assuming SW11=1 (DELETE ITERATIONS) is set to give the fastest possible execution. The actual execution time depends greatly on the PDP11 CPU configuration and the amount of memory in the system.

8.2 PASS COMPLETE

NOTE: EVERY time the program is started; the tests will run as if SW11 (delete iterations) was up (=1). This is to 'VERIFY NO HARD ERRORS' as soon as possible. Therefore the first pass -EACH TIME PROGRAM IS STARTED- will be a 'QUICK PASS' until all KMC11's in system are tested. When the diagnostic has completed a pass the following is an example of the print out to be expected.

```
END PASS DZKCA CSR:  175000 VEC:  0300 PASSES:  000001
ERRORS:  000000
```

NOTE: The pass count and error counts are cummulitive for each KMC11 that is running, and are set to zero only when the diagnostic is started. Therefore after an overnight run for example, the total passes and errors for each KMC11 since the diagnostic was started are reflected in PASSES: and ERRORS:.

8.4 KEY LOCATIONS

SLPADR (1206) Contains the address where program will return when iteration count is reached or if loop on test is asserted.

NEXT (1442) Contains the address of the next test to be peformed.

STSTNM (1202) Contains the number of the test now being peformed.

RUN (1500) The bit in 'RUN' always points to the KMC11 currently being tested. EXAMPLE: (RUN) 1500/0000000001000000 Means that KMC11 no.06 is the KMC11 now running.

KMCR00-KMCR17
KMST00-KMST17
(2100)-(2300)

These locations contain the information needed to test up to 16 (decimal) KMC11s sequentialy. they contain the CSR,VECTOR and STATUS concerning the configuration of each KMC11.

KMACTV (1306) Each bit set in this location indicates that the associated KMC11 will be tested in turn. EXAMPLE: (KMACTV) 1470/0000000000011111 means that KMC11 no. 00,01,02,03,04 will be tested. EXAMPLE: (KMACTV) 1470/0000000000010001 Means that KMC11 no. 00,04 will be tested.

KMCSR (2066) Contains the CSR of the current KMC11 under test.

8.4A 'STATUS TABLE' (2100-2300)

The table is filled by AUTO SIZING or by the manual parameter input (questions) as described previously. Also if desired by user; the locations may be altered by hand (toggled in) to suit the specific configuration.

The example status map shown below contains information for two KMC11'S. the table can contain up to 16 KMC11'S. Following the map is a description of the bits for each map entry

MAP OF KMC11 STATUS

PC	CSR	STAT1	STAT2	STAT3
002100	160010	045310	177777	000000
002110	160020	016320	000000	000000

Each map entry contains 4 words which contain the status information for 1 KMC11. The PC shows where in core memory the first of the 4 words is. In the example above the first KMC'S status is in locations, 2100, 2102, 2104, and 2106. The second KMC status is located at 2110, 2112, 2114, and 2116. The information contained in each 4 word entry is defined as follows:

CSR: Contains KMC11 CSR address

STAT1: BITS 00-08 IS KMC11 VECTOR ADDRESS
BIT14=1 TURNAROUND CONNECTOR IS ON
BIT14=0 NO TURNAROUND CONNECTOR
BIT13=0 LINE UNIT IS AN MB201
BIT13=1 LINE UNIT IS AN MB202
BIT12=1 NO LINE UNIT
BITS 09-11 IS KMC11 BR PRIORITY LEVEL

STAT2: LOW BYTE IS SWITCH PAC#1 (DDCMP LINE NUMBER)
HIGH BYTE IS SWITCH PAC#2 (BM873 BOOT ADD)

STAT3: NOT USED

8.5 METHOD OF AUTO SIZING

8.5.1 FINDING THE CONTROL STATUS REGISTER.

The auto-sizing routine finds a KMC11 as follows: It starts at address 160000 and tests all address in increments of 10 up to and including address 167760. If the address does not time out, the following is done, the first CRAM address is written to a 125252 then it is read back. If it contains a -1 or 125252 a KMC11 has been found, if not, the address is updated by 10 and the search continues. A -1 indicates a KMC11 with no CRAM, and a 125252 indicates a KMC11 with CRAM. Further tests are performed at this point to determine which line unit, if any, is installed, if a loop-back connector is installed and various switch settings on the line unit. THIS IS WHY THE STATUS TABLE MUST BE VERIFIED BY THE USER AND IF ANY OF THE INFORMATION DOES NOT AGREE WITH THE HARDWARE THE DIAGNOSTIC MUST BE RESTARTED AND THE QUESTIONS MUST BE ANSWERED. All KMC11's in the system will be found by the auto-sizer. If it does not find a KMC11 the diagnostic must be restarted and the questions answered.

8.5.2 FINDING THE VECTOR AND BR LEVEL

The vector area (address 300-776) is filled with the instruction IOT and '.+2' (next address). The processor status is started at 7 and the KMC is programmed to interrupt. The PS is lowered by 1 until the KMC interrupts, a delay is made and if no interupt occures at PS level 3 (because of a bad KMC11) the program assumes vector address 300 at BR level 5 and the problem should be fixed in the diagnostic. Once the problem is fixed; the program should be re-setup again to get correct vector. If an interupt occured; the address to which the KMC11 interupted to is picked up and reported as the vector. NOTE: if the vector reported is not the vector set up by you; there is a problem and AUTO SIZING should not be done.

8.5 SOFTWARE SWITCH REGISTER

If the diagnostic is run on an 11/04 or other CPU without a switch register then a software switch register is used to allow user the same switch options as described previously. If the hardware switch register does not exist or if one does and it contains all ones (177777) this software switch register is used.

Control:

To obtain control at any allowable time during execution of the diagnostic the operator types a CTRL G on the console terminal keyboard. As soon as the CTRL G is recognized, by the diagnostic, the following message will be displayed:

SWR=XXXXXX NEW?

Where XXXXXX is the current contents of the software switch register in octal. The software control routine will then await operator action. At which time the operator is required to type one or more of the legal characters: 1) 0 - 7, 2) line feed(<LF>), 3) carriage return(<CR>), or 4) control-U (CTRL U). No check is made for legality. If the input character is not a <LF>, <CR>, or CTRL U it is assumed to be an octal digit.

To change the contents of the SSR the operator simply types the new desired value in octal - leading zeros need not be typed. And terminates the input string with a <CR> or <LF> depending on the program action desired as described below. The input value will be truncated to the last 6 digits typed. At least one digit must be typed on any given input string prior to the terminator before a change to the SSR will occur.

When the input string is terminated with a <CR> the diagnostic will continue execution from the point at which it was interrupted. If a <CR> is the only thing typed the program will continue without changing the SSR. The <LF> differs from the <CR> by restarting the program as if it were restarted at address 200.

If a CTRL U is typed at any point in the input string prior to the terminator the input value will be disregarded and the prompt displayed (SWR = XXXXXX NEW?).

To set the SSR for the starting switches, first load the diagnostic, then hit CTRL G, then start the diagnostic.

APT/ACT/XXDP/SLIDE

++++++++++++++++++++++++++

THIS DIAGNOSTIC IS APT/ACT/XXDP/SLIDE COMPATIBLE USER WOULD BE ABLE TO RUN IT UNDER APT/ACT/XXDP ENVIRONMENT.

NOTE: FOR MANUFACTURING PURPOSE ONLY ITS DESCRIBED HOW TO RUN UNDER APT ENVIRONMENT.

##

ETABLE SETTING FOR APT TO RUN UNDER APT

++

FIRST PASS TIME:

LONGEST TEST TIME:

ADDITIONAL TEST TIME:

ALL THE ABOVE PARAMETERS ARE DEPENDENT ON PARTICULAR DIAGNOSTICS AND SHOULD BE LOADED AT THE TIME OF SETTING ETABLE.THERE IS NO DEFAULT TIME SET UP.

SOFTWARE ENVIRONMENT:001 ENVIRONMENT MODE:200

SWITCH 1:-SHOULD BE USED AS NORMAL SWITCH REGISTER.

SWITCH 2:-NOT USED.

CPU OPTIONS:-NOT USED.

MEMORY TYPE 1:-BITS<2:4>:=BITS <12:14> OF STAT1 OF DEV:0.

MAXIMUM ADDRESS:-BITS<17:19>:=BITS<12:14> OF STAT1 OF DEV:1
BITS<2:4>:=BITS <12:14> OF STAT1 OF DEV:2
BITS<10:12>:=BITS<12:14> OF STAT1 OF DEV:3

IN THE SAME MANNER

MEMORY TYPE 2 MAXIMUM ADDRESS:-GETS STAT1<12:14> OF DEVICE 4,5,6,7.

MEMORY TYPE 3 MAXIMUM ADDRESS:-GETS STAT1<12:14> OF DEVICE 8,9,10,11.

MEMORY TYPE 4 MAXIMUM ADDRESS:-GETS STAT1<12:14> OF DEVICE 12,13,14,15.

INTERRUPT VECTOR 1:FIRST DEVICE RECEIVE VECTOR.

REST OF THE DEVICE(KMC'S) VECTOR SHOULD BE SET UP SEQUENTIALLY IN INCREMENTS OF 10.

BUS PRIORITY:KMC'S PRIORITY(SHOULD BE SAME FOR ALL KMC'S UNDER TEST).

INTERRUPT VECTOR 2:NOT USED.

BUS PRIORITY:NOT USED.

BASE ADDRESS:FIRST DEVICE CSR ADDRESS.

REST SHOULD FOLLOW SEQUENTIALLY

IN INCREMENTS OF 10.

DEVICE MAP:AS DESCRIBED IN APT MANUAL.

CONTROLLER SPECIFIC CODE 1:-NO. OF DEVICES UNDER TEST.

CONTROLLER SPECIFIC CODE 2:-NOT USED.

DEVICE DESCRIPTOR WORD 0:STAT2 OF FIRST DEVICE.

. .

. .

. .

TO

. .

. .

. .

DEVICE DESCRIPTOR WORD 15:STAT2 OF 16TH DEVICE.(KMC)

****

FOR KMC11 CPU MICRO-DIAGNOSTICS ONLY ::

++ THIS DIAGNOSTIC TESTS ARE STRUCTURED IN FOLLOWING MANNER

1. SAVE CURRENT TEST NO AND ADDRESS OF NEXT TEST AS USUALLY DONE.

2. LOAD THE MICRO-CODE FOR THE PARTICULAR TEST.VERIFY WHAT WAS LOADED.AND THEN WAIT FOR THE TEST TO INTERRUPT IN SPECIFIC TIME.

3. IF THE TEST DOES NOT INTERRUPT IN SPECIFIC TIME [IN WHICH IT SHOULD HAVE COMPLETED] THEN FLAG IT AS TIME OUT ERROR AND START THE NEXT TEST.(NO LOOP ON ERROR OR SCOPE ON THIS CONDITION.)

4. IF THE MICRO-PROCESSOR PC LOGIC HAS FAULT AND IT GOES OUT OF TEST BOUNDRY THEN "UPC[MICRO-PROCESSOR PC] SEQUENCE ERROR " WOULD BE REPORTED.HERE AGAIN LOOP ON ERROR IS NOT SUPPORTED HOWEVER DIAGNOSTIC SCOPE FEATURE IS SUPPORTED. WHEN DOING NEXT ITERATION IT WOULD START THIS TEST FROM BEGINING.

5. IF ERROR ON LOADING MICRO-CODE, IT WOULD REPORT ERROR AND HALT- FOR FURTHER ACTION FROM OPERATOR.

AT THIS POINT

STSTNM=WILL HAVE THE TEST NO. AT WHICH ERROR OCCURED.

R0=POINTS TO GOOD MICRO-CODE INSTRUCTION.

R1=HAS CSR ADDRESS OF KMC UNDER TEST.

R2=POINTS THE CRAM LOCATION.

6. ALL THE REGULAR DATA ERRORS AND HARDWARE ERRORS ARE REPORTED WITH

TEST NO: XXX PC:STARTING ADDRESS OF THE TEST.

" ERROR MESSAGE "

GOOD BAD UPC XXXXXX[IF REQUIRED]

WITH ABOVE INFORMATION USER CAN LOCATE ERROR OCCURENCE IN PROGRAM LISTING AT TEST NO: XXX OR PC=START ADDRESS OF THE TEST. THEN LOOKING FOR MICRO-PROCESSOR PC[MICPC]=UPC.

****
*

DOCUMENT
****
MAINDEC-11-DZKCA-A
****

COPYRIGHT 1977
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

```
2262    ************************** TEST 1 **************************
        MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
        FLOAT A 1 THROUGH REGISTER OUT1 (4)
        FLOAT A 0THROUGH REGISTER OUT1 (4)

2459    ************************** TEST 2 **************************
        MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
        FLOAT A 1 THROUGH REGISTER OUT1 (5)
        FLOAT A 0THROUGH REGISTER OUT1 (5)

2656    ************************** TEST 3 **************************
        MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
        FLOAT A 1 THROUGH REGISTER OUT1 (6)
        FLOAT A 0THROUGH REGISTER OUT1 (6)

2853    ************************** TEST 4 **************************
        MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
        FLOAT A 1 THROUGH REGISTER OUT1 (7)
        FLOAT A 0THROUGH REGISTER OUT1 (7)

3050    ************************** TEST 5 **************************
        MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
        FLOAT A 1 THROUGH REGISTER OUT1 (10)
        FLOAT A 0THROUGH REGISTER OUT1 (10)
        THE NPR RQ BIT (BIT0) IS MASKED DURING THIS TEST.

3266    ************************** TEST 6 **************************
        MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
        FLOAT A 1 THROUGH REGISTER OUT1 (11)
        FLOAT A 0THROUGH REGISTER OUT1 (11)
        THE BR RQ BIT, PGM CLOCK BIT, FORCE POWER FAIL BIT
        (BITS 7,4,1) ARE ALL MASKED DURING THIS TEST

3501    ************************** TEST 7 **************************
        MICRO PROCESSOR OUT0 REGISTER WRITE/READ TEST.
        FLOAT A 1 THROUGH REGISTER OUT0 (0)
        FLOAT A 0THROUGH REGISTER OUT0 (0)
```

```
3698    ************************** TEST 10 **************************
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.

3700     FLOAT A 1 THROUGH REGISTER OUTO <1>
         FLOAT A 0THROUGH REGISTER OUTO <1>

3895    ************************** TEST 11 **************************
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUTO <2>
         FLOAT A 0THROUGH REGISTER OUTO <2>

4092    ************************** TEST 12 **************************
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUTO <3>
         FLOAT A 0THROUGH REGISTER OUTO <3>

4289    ************************** TEST 13 **************************
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUTO <4>
         FLOAT A 0THROUGH REGISTER OUTO <4>

4486    ************************** TEST 14 **************************
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUTO <5>
         FLOAT A 0THROUGH REGISTER OUTO <5>

4683    ************************** TEST 15 **************************
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUTO <6>
         FLOAT A 0THROUGH REGISTER OUTO <6>

4880    ************************** TEST 16 **************************
         MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
         FLOAT A 1 THROUGH REGISTER OUTO <7>
         FLOAT A 0THROUGH REGISTER OUTO <7>

5077    ************************** TEST 17 **************************
         MICRO PROCESSOR 8 REGISTER TEST
         FLOAT A 1 THROUGH THE BREG.
         FLOAT A 0 THROUGH THE BREG.

5255    ************************** TEST 20 **************************
         SCRATCH PAD TEST FOR SP4
         FLOAT A 1 THROUGH SCRATCH PAD 4
         FLOAT A 0 THROUGH SCRATCH PAD 4

5462    ************************** TEST 21 **************************
         SCRATCH PAD TEST FOR SP5
         FLOAT A 1 THROUGH SCRATCH PAD 5
         FLOAT A 0 THROUGH SCRATCH PAD 5
```

```
5669    ********** TEST 22 **********
        SCRATCH PAD TEST FOR SP6
        FLOAT A 1 THROUGH SCRATCH PAD 6
        FLOAT A 0 THROUGH SCRATCH PAD 6

5876    ********** TEST 23 **********
        SCRATCH PAD TEST FOR SP7
        FLOAT A 1 THROUGH SCRATCH PAD 7
        FLOAT A 0 THROUGH SCRATCH PAD 7

6083    ********** TEST 24 **********
        SCRATCH PAD TEST FOR SP10
        FLOAT A 1 THROUGH SCRATCH PAD 10
        FLOAT A 0 THROUGH SCRATCH PAD 10

6290    ********** TEST 25 **********
        SCRATCH PAD TEST FOR SP11
        FLOAT A 1 THROUGH SCRATCH PAD 11
        FLOAT A 0 THROUGH SCRATCH PAD 11

6497    ********** TEST 26 **********
        SCRATCH PAD TEST FOR SP12
        FLOAT A 1 THROUGH SCRATCH PAD 12

6500    FLOAT A 0 THROUGH SCRATCH PAD 12

6704    ********** TEST 27 **********
        SCRATCH PAD TEST FOR SP13
        FLOAT A 1 THROUGH SCRATCH PAD 13
        FLOAT A 0 THROUGH SCRATCH PAD 13

6911    ********** TEST 30 **********
        SCRATCH PAD TEST FOR SP14
        FLOAT A 1 THROUGH SCRATCH PAD 14
        FLOAT A 0 THROUGH SCRATCH PAD 14

7118    ********** TEST 31 **********
        SCRATCH PAD TEST FOR SP15
        FLOAT A 1 THROUGH SCRATCH PAD 15
        FLOAT A 0 THROUGH SCRATCH PAD 15

7325    ********** TEST 32 **********
        NPR TEST
        TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC

7498    ********** TEST 33 **********
        NPR TEST
        TEST OF DATO, 1 WORD FROM UPROC TO PDP11 MEMORY.
        THEN DATI OF THAT WORD AND CHECK....
```

```
7741    ********************** TEST 34 **********************
        NPR NON-EXISTENT MEMORY TEST
        DO A DATO TO A NON-EXISTENT ADDRESS.
        VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11

7877    ********************** TEST 35 **********************
        NPR TEST
        TEST OF MULTIPLE NPR'S DOING DATI.
        XFER 6 WORD'S FROM 11 MEMORY TO UPROC.


8119    ********************** TEST 36 **********************
         MAIN MEMORY TEST
         FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATION...

8276    ********************** TEST 37 **********************
         MAIN MEMORY TEST
         FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS...

8436    ********************** TEST 40 **********************
         MAIN MEMORY DUAL ADDRESSING TEST
         LOAD EACH MEMORY LOCATION WITH IT'S OWN PAGE LESS ADDRESS.
         READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING.

8649    ********************** TEST 41 **********************
         MAR TEST.
         PERFORM DUAL ADDRESSING TEST.
         USING MAR AUTO-INC FEATURE.

8815    ********************** TEST 42 **********************
        ALU C BIT TEST
        TEST THAT ADD OF 377 AND 1 WILL SET TAG CBIT.
        THEN CHECK IF C BIT CLEARS

8938    ********************** TEST 43 **********************
        ALU TEST
        TEST OF ALU FUNCTION SEL B & SEL A WITH C BIT CLEARED.
        TEST OF ALU FUNCTION SEL B & SEL A WITH C BIT SET.
        ALU FUNCTION (B)
        LOAD MAIN MEMORY 16 WORDS OF DATA.
        PERFORM THE FUNCTION, VERIFY THE RESULTS..

9255    ********************** TEST 44 **********************
        ALU TEST
        TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED.
        TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET.
        ALU FUNCTION (A OR NOTB)
        LOAD MAIN MEMORY 16 WORDS OF DATA.
        PERFORM THE FUNCTION, VERIFY THE RESULTS..
```

```
9572    ************************** TEST 45 ****************************
        ALU TEST
        TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED.
        TEST OF ALU FUNCTION A AND B WITH C BIT SET.
        ALU FUNCTION (A AND B)
        LOAD MAIN MEMORY 16 WORDS OF DATA.
        PERFORM THE FUNCTION, VERIFY THE RESULTS..

9889    ************************** TEST 46 ****************************
        ALU TEST
        TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED.
        TEST OF ALU FUNCTION A OR B WITH C BIT SET.
        ALU FUNCTION (A OR B)
        LOAD MAIN MEMORY 16 WORDS OF DATA.
        PERFORM THE FUNCTION, VERIFY THE RESULTS..

10206   ************************** TEST 47 ****************************
        ALU TEST
        TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED.
        TEST OF ALU FUNCTION A XOR B WITH C BIT SET.
        ALU FUNCTION (A XOR B)
        LOAD MAIN MEMORY 16 WORDS OF DATA.
        PERFORM THE FUNCTION, VERIFY THE RESULTS..

10523   ************************** TEST 50 ****************************
        ALU TEST
        TEST OF ALU FUNCTION ADD WITH C BIT CLEARED.
        TEST OF ALU FUNCTION ADD WITH C BIT SET.
        ALU FUNCTION (A PLUS B)
        LOAD MAIN MEMORY 16 WORDS OF DATA.
        PERFORM THE FUNCTION, VERIFY THE RESULTS..

10840   ************************** TEST 51 ****************************
        ALU TEST
        TEST OF ALU FUNCTION 2A W/C WITH C BIT CLEARED.
        TEST OF ALU FUNCTION 2A W/C WITH C BIT SET.
        ALU FUNCTION (A PLUS A PLUS C)
        LOAD MAIN MEMORY 16 WORDS OF DATA.
        PERFORM THE FUNCTION, VERIFY THE RESULTS..

11157   ************************** TEST 52 ****************************
        ALU TEST
        TEST OF ALU FUNCTION SUB WITH C BIT CLEARED.
        TEST OF ALU FUNCTION SUB WITH C BIT SET.
        ALU FUNCTION (A-B)
        LOAD MAIN MEMORY 16 WORDS OF DATA.
        PERFORM THE FUNCTION, VERIFY THE RESULTS..

11474   ************************** TEST 53 ****************************
        ALU TEST
        TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED.
        TEST OF ALU FUNCTION ADD W/C WITH C BIT SET.
        ALU FUNCTION (A PLUS B PLUS C)
        LOAD MAIN MEMORY 16 WORDS OF DATA.
```

```
                PERFORM THE FUNCTION, VERIFY THE RESULTS..

11791   ********************************* TEST 54 *****************************
                ALU TEST
                TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED.
                TEST OF ALU FUNCTION SUB W/C WITH C BIT SET.
                ALU FUNCTION (A-B-C)
                LOAD MAIN MEMORY 16 WORDS OF DATA.
                PERFORM THE FUNCTION, VERIFY THE RESULTS..

12108   ********************************* TEST 55 *****************************
                ALU TEST
                TEST OF ALU FUNCTION INC A WITH C BIT CLEARED.
                TEST OF ALU FUNCTION INC A WITH C BIT SET.
                ALU FUNCTION (A PLUS 1)
                LOAD MAIN MEMORY 16 WORDS OF DATA.
                PERFORM THE FUNCTION, VERIFY THE RESULTS..

12425   ********************************* TEST 56 *****************************
                ALU TEST
                TEST OF ALU FUNCTION 2A WITH C BIT CLEARED.
                TEST OF ALU FUNCTION 2A WITH C BIT SET.
                ALU FUNCTION (A PLUS A)
                LOAD MAIN MEMORY 16 WORDS OF DATA.
                PERFORM THE FUNCTION, VERIFY THE RESULTS..

12742   ********************************* TEST 57 *****************************
                ALU TEST
                TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED.
                TEST OF ALU FUNCTION A PLUS C WITH C BIT SET.
                ALU FUNCTION (A PLUS C)
                LOAD MAIN MEMORY 16 WORDS OF DATA.
                PERFORM THE FUNCTION, VERIFY THE RESULTS..

13059   ********************************* TEST 60 *****************************
                ALU TEST
                TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED.
                TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET.
                ALU FUNCTION (A-B-1)
                LOAD MAIN MEMORY 16 WORDS OF DATA.
                PERFORM THE FUNCTION, VERIFY THE RESULTS..

13376   ********************************* TEST 61 *****************************
                ALU TEST
                TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED.
                TEST OF ALU FUNCTION DEC A WITH C BIT SET.
                ALU FUNCTION (A-1)
                LOAD MAIN MEMORY 16 WORDS OF DATA.
                PERFORM THE FUNCTION, VERIFY THE RESULTS..
```

```
13693   ************************** TEST 62 ****************************
        TEST OF PROGRAM CLOCK BIT
        DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
        WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS.
        AND THEN SETS SOME TIME LATER.

13834   ************************** TEST 63 ****************************
        MICRO-PROCESSOR NOISE TEST.

13836   WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN TO
        THE IBUS#,IBUS,SP & MAIN MEMORY
        THEN GO AND READ BACK THE DATA PATTERN
        TO VERIFY THAT READING AND WRITING OF OTHER
        LOCATIONS DID NOT CHANGE DATA.

14461   ************************** TEST 64 ****************************
         HELL RAISER TEST...
         ONLY TO TEST MPRAM CONTROL LOGIC...
         NOT FOR MAINTENENCE PURPOSE...

14677   ************************** TEST 65 ****************************
        FORCE POWER FAIL TEST.
        SET FORCE POWER FAIL VERIFY THAT PROCESSOR TRAPS TO LOC 24.
        GOING DOWN AND COMING UP. VERIFY ALSO THAT BUS INIT WAS
        BLOCKED FROM GETTING TO KMC DURING THE POWER FAIL .

14797   ***************************************************************************

        ERROR REPORT ROUTINE. INTERRUPTS AT LOCATION XXO.
        1) CSR4:=GOOD DATA      2) CSR5:=BAD DATA
        3) CSR3:=ERROR TYPE     4) CSR6:=ERROR PC IN MICRO-CODE.
        5) CSR7:=MISCALLANEOUS INFORMATION.

       ***************************************************************************
```

```
                        KMERNT  ↑?MAINDEC-11-DZKCA-A? DZKCA ↑?BASIC KMC11 CONTROLLER TEST?
                        .HEADER <MAINDEC-11-DZKCA-A>,1976,<DINESH GORADIA>
                        .TITLE  MAINDEC-11-DZKCA-A
                        ;*COPYRIGHT (C) 1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY DINESH GORADIA
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*
000002                  $HEADER <1976>,↑?MAINDEC-11-DZKCA-A?,<DZKCA>,<KMC11>,↑?BASIC KMC11 CONTROLLER TEST?



                        ;*MAINDEC-11-DZKCA-A   BASIC KMC11 CONTROLLER TEST
                        ;*COPYRIGHT 1976, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
                        ;*-----------------------------------------------------------

                        ;STARTING PROCEDURE
                        ;LOAD PROGRAM
                        ;LOAD ADDRESS 000200
                        ;SW8=0  AUTOSIZE KMC11
                        ;SW07=1  USE CURRENT KMC11 PARAMETERS
                        ;SW00=1  INPUT NEW KMC11 PARAMETERS
                        ;PRESS START
                        ;PROGRAM WILL TYPE "MAINDEC-11-DZKCA-A   BASIC KMC11 CONTROLLER TEST"
                        ;PROGRAM WILL TYPE STATUS MAP
                        ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
                        ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                        ;AND THEN RESUME TESTING
                        ;SUBSEQUENT RESTARTS WILL NOT TYPE PROGRAM TITLE

                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1200 ***
001200                  STACK=  1200
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
```

```
                        ;*GENERAL REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
```

```
                        BIT09=  1000
                        BIT08=  400
                        BIT07=  200
                        BIT06=  100
                        BIT05=  40
                        BIT04=  20
                        BIT03=  10
                        BIT02=  4
                        BIT01=  2
                        BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PURVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR


                        ;INSTRUCTION DEFINITIONS
                        ;-----------------------

005746                  PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
005726                  POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
010046                  PUSHR0=10046    ;SAVE R0 ON STACK
012600                  POPR0=12600     ;RESTORE R0 FROM STACK
024646                  PUSH2SP=24646   ;DECREMENT STACK TWICE
022626                  POP2SP=22626    ;INCREMENT STACK TWICE
                        .EQUIV  EMT,HLT ;BASIC DEFINITION OF ERROR CALL
```

```
                                ;:*****************************************************************************
                                ;-----------------------------------------------------------------------------
                                ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
                                ;THE STANDARD "TRAP CATCHER" IS PLACED
                                ;BETWEEN ADDRESS 0 TO ADDRESS 776.
                                ;IT LOOKS LIKE "PC+2 HALT".
                                ;-----------------------------------------------------------------------------
                                ;:*****************************************************************************

            000000              .=0
000000  000000  000000                .WORD   0,0
                                ;STANDARD INTERRUPT VECTORS
                                ;-------------------------

            000020              .=20
000020  004134                      $SCOPE          ; SCOPE LOOP HANDLER.
000022  000340                      PR7             ; SERVICE AT LEVEL 7.
000024  007100                      $PWRDN                  ;POWER FAIL HANDLER
000026  000340                      PR7                     ;SERVICE AT LEVEL 7
000030  006502                      $ERROR                  ;ERROR HANDLER
000032  000340                      PR7                     ;SERVICE AT LEVEL 7
000034  006404                      $TRAP                   ;GENERAL HANDLER DISPATCH SERVICE
000036  000340                      PR7                     ;SERVICE AT LEVEL 7
                                .SBTTL  ACT11 HOOKS

                                ;;*****************************************************************************
                                ;HOOKS REQUIRED BY ACT11
            000040                      $SVPC=.                 ;SAVE PC
            000046                      .=46
000046  004070                      $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
            000052                      .=52
000052  000000                      .WORD   0               ;;2)SET LOC.52 TO ZERO
            000040                      .=$SVPC                 ;; RESTORE PC

            000174              .=174
000174  000000              DISPREG:0                       ;SOFTWARE DISPLAY REGISTER
000176  000000              SWREG:  0                       ;SOFTWARE SWITCH REGISTER

            000200              .=200
000200  000137  002402              JMP     .START          ;GO TO START OF PROGRAM

            001000              .=1000
001000  005200  040515  047111  MTITLE: .ASCII  <200><12>/MAINDEC-11-DZKCA-A/<200>
(2) 001025  102  051501  041511          .ASCIZ  /BASIC KMC11 CONTROLLER TEST/<200>
(2)
            177570              DSWR    =       177570
            177570              DDISP   =       177570
```

```
                        .SBTTL  COMMON TAGS

                        ;;**********************************************************************
                        ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                        ;*USED IN THE PROGRAM.

                001200          .=1200
001200                  SCMTAG:                         ;;START OF COMMON TAGS
001200  000000                  .WORD   0
001202  000          STSTNM: .BYTE   0                  ;;CONTAINS THE TEST NUMBER
001203  000          SERFLG: .BYTE   0                  ;;CONTAINS ERROR FLAG
001204  000000       SICNT:  .WORD   0                  ;;CONTAINS SUBTEST ITERATION COUNT
001206  000000       SLPADR: .WORD   0                  ;;CONTAINS SCOPE LOOP ADDRESS
001210  000000       SLPERR: .WORD   0                  ;;CONTAINS SCOPE RETURN FOR ERRORS
001212  000000       SERTTL: .WORD   0                  ;;CONTAINS TOTAL ERRORS DETECTED
001214  000          SITEMB: .BYTE   0                  ;;CONTAINS ITEM CONTROL BYTE
001215  001          SERMAX: .BYTE   1                  ;;CONTAINS MAX. ERRORS PER TEST
001216  000000       SERRPC: .WORD   0                  ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001220  000000       SGDADR: .WORD   0                  ;;CONTAINS ADDRESS OF 'GOOD' DATA
001222  000000       SBDADR: .WORD   0                  ;;CONTAINS ADDRESS OF 'BAD' DATA
001224  000000       SGDDAT: .WORD   0                  ;;CONTAINS 'GOOD' DATA
001226  000000       SBDDAT: .WORD   0                  ;;CONTAINS 'BAD' DATA
001230  000000               .WORD   0                  ;;RESERVED--NOT TO BE USED
001232  000000               .WORD   0
001234  000          SAUTOB: .BYTE   0                  ;;AUTOMATIC MODE INDICATOR
001235  000          SINTAG: .BYTE   0                  ;;INTERRUPT MODE INDICATOR
001236  000000               .WORD   0
001240  177570       SWR:    .WORD   DSWR               ;;ADDRESS OF SWITCH REGISTER
001242  177570       DISPLAY: .WORD  DDISP              ;;ADDRESS OF DISPLAY REGISTER
                     STKS:   177560                     ;;TTY KBD STATUS
                     STKB:   177562                     ;;TTY KBD BUFFER
                     STPS:   177564                     ;;TTY PRINTER STATUS REG. ADDRESS
                     STPB:   177566                     ;;TTY PRINTER BUFFER REG. ADDRESS
001244  000          SNULL:  .BYTE   0                  ;;CONTAINS NULL CHARACTER FOR FILLS
001245  002          SFILLS: .BYTE   2                  ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001246  012          SFILLC: .BYTE   12                 ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001247  000          STPFLG: .BYTE   0                  ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001250  000000       SREGAD: .WORD   0                  ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  (SREG0) WAS OBTAINED
001252  000000       SREG0:  .WORD   0                  ;;CONTAINS ((SREGAD)+0)
001254  000000       SREG1:  .WORD   0                  ;;CONTAINS ((SREGAD)+2)
001256  000000       SREG2:  .WORD   0                  ;;CONTAINS ((SREGAD)+4)
001260  000000       SREG3:  .WORD   0                  ;;CONTAINS ((SREGAD)+6)
001262  000000       SREG4:  .WORD   0                  ;;CONTAINS ((SREGAD)+10)
001264  000000       SREG5:  .WORD   0                  ;;CONTAINS ((SREGAD)+12)
001266  000000       STMP0:  .WORD   0                  ;;USER DEFINED
001270  000000       STMP1:  .WORD   0                  ;;USER DEFINED
001272  000000       STMP2:  .WORD   0                  ;;USER DEFINED
001274  000000       STMP3:  .WORD   0                  ;;USER DEFINED
001276  000000       STMP4:  .WORD   0                  ;;USER DEFINED
001300  000000       STIMES: 0                          ;;MAX. NUMBER OF ITERATIONS
001312  077          $QUES:  .ASCII  /?/                ;;QUESTION MARK
001313  015          $CRLF:  .ASCII  <15>               ;;CARRIAGE RETURN
001314  000012       $LF:    .ASCIZ  <12>               ;;LINE FEED
                        ;;**********************************************************************
                        .SBTTL  APT MAILBOX-ETABLE
```

```
                                ;;*****************************************************************
                                .EVEN
                                SMAIL:
001316                          SMSGTY: .WORD   AMSGTY  ;;APT MAILBOX
001316  000000                  SFATAL: .WORD   AFATAL  ;;MESSAGE TYPE CODE
001320  000000                  STESTN: .WORD   ATESTN  ;;FATAL ERROR NUMBER
001322  000000                  SPASS:  .WORD   APASS   ;;TEST NUMBER
001324  000000                  SDEVCT: .WORD   ADEVCT  ;;PASS COUNT
001326  000000                  SUNIT:  .WORD   AUNIT   ;;DEVICE COUNT
001330  000000                  SMSGAD: .WORD   AMSGAD  ;;I/O UNIT NUMBER
001332  000000                  SMSGLG: .WORD   AMSGLG  ;;MESSAGE ADDRESS
001334  000000                  SETABLE:                ;;MESSAGE LENGTH
001336                          SENV:   .BYTE   AENV    ;;APT ENVIRONMENT TABLE
001336      002                 SENVM:  .BYTE   AENVM   ;;ENVIRONMENT BYTE
001337      000                 SSWREG: .WORD   ASWREG  ;;ENVIRONMENT MODE BITS
001340  000000                  SUSWR:  .WORD   AUSWR   ;;APT SWITCH REGISTER
001342  000000                  SCPUOP: .WORD   ACPUOP  ;;USER SWITCHES
001344  000000                                          ;;CPU TYPE,OPTIONS
                                                        BITS 15-11=CPU TYPE
                                                                11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                                                11/70=06,PDP=07,0=10
                                                        BIT 10=REAL TIME CLOCK
                                                        BIT  9=FLOATING POINT PROCESSOR
                                                        BIT  8=MEMORY MANAGEMENT
001346      000                 SMAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
001347      000                 SMTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                                                        MEM.TYPE BYTE   --  (HIGH BYTE)
                                                                000 NSEC CORE=001
                                                                000 NSEC BIPOLAR=002
                                                                000 NSEC MOS=003
001350  000000                  SMADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                                                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001352      000                 SMAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
001353      000                 SMTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
001354  000000                  SMADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
001356      000                 SMAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
001357      000                 SMTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
001360  000000                  SMADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
001362      000                 SMAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
001363      000                 SMTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
001364  000000                  SMADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
001366  000000                  SVECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001370  000000                  SVECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001372  000000                  SBASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001374  000000                  SDEVM:  .WORD   ADEVM   ;;DEVICE MAP
001376  000000                  SCDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
001400  000000                  SCDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
001402  000000                  SDDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
001404  000000                  SDDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
001406  000000                  SDDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
001410  000000                  SDDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
001412  000000                  SDDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
001414  000000                  SDDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
001416  000000                  SDDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
001420  000000                  SDDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
001422  000000                  SDDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
```

```
001424  000000          SDDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
001426  000000          SDDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
001430  000000          SDDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
001432  000000          SDDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
001434  000000          SDDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
001436  000000          SDDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
001440  000000          SDDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15

001442                  SETEND:

                        ;
                        ;  PROGRAM CONTROL PARAMETERS
                        ;-------------------------------
001442  000000          NEXT:   .WORD   0               ; ADDRSS OF NEXT TEST TO BE EXECUTED
001444  000000          LOCK:   .WORD   0               ; ADDRESS FOR LOCK CURRENT DATA
                        ;
                        ;   PROGRAM VARIABLES
                        ;-------------------------
001446  000000          STRTSW: .WORD   0               ; SWITCHES AT START OF PROGRAM
001450  000000          STAT:   .WORD   0               ; KM STATUS WORD STORAGE
001452  000000          CLICX:  .WORD   0               ;
001454  000000          MASKX:  .WORD   0               ;
001456  000000          SAVSP:  .WORD   0               ; STACK POINTER STORAGE
001460  000000          SAVPC:  .WORD   0               ; PROGRAM COUNTER STORAGE
001462  000000          ZERO:   .WORD   0               ;
001464  000001          ONE:    .WORD   1               ;
001466  000000          MEMLIM: .WORD   0               ; HIGHEST LOCATION FOR NPR'S
001470  000001          KMACTV: .BLKW   1               ; KMC11 SELECTED ACTIVE
001472  000001          KMNUM:  .BLKW   1               ; OCTAL NUMBER OF KMC11'S
001474  000001          SAVACT: .BLKW   1               ; ORIGINAL ACTIVE DEVICES.
001476  000001          SAVNUM: .BLKW   1               ; WORKABLE NUMBER.
001500  000000          RUN:    .WORD   0               ; POINTER TO RUNNING DEVICES
                                .EVEN
001502  002072          CREAM:  .WORD   KM.MAP-6        ; TABLE POINTER
001504  002276          MILK:   .WORD   CNT.MAP-4       ; TABLE POINTER
                        ;
                        ;  PROGRAM CONTROL FLAGS
                        ;-----------------------
001506  000           INIFLG: .BYTE   0               ; PROGRAM INITIALIZING FLAG
001510                        .EVEN
001510  000           LOKFLG: .BYTE   0               ; LOCK ON CURRENT TEST FLAG
001511  000           QV.FLG: .BYTE   0               ; QUICK VERIFY FLAG
                                                      ; ON FIRST PASS OF EACH KMC11 ITERATIONS WILL BE SUPPRES

                                .EVEN
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT

001512                          SERRTB:
                                .EVEN
                                ;*      DF              ;; DOES NOT APPLY IN THIS DIAGNOSTIC.
001512  000000                          0
001514  000000                          0
001516  000000                          0
001520  036176                  EM1
001522  037256                  DH1             ; ERROR 1
001524  037434                  DT1
001526  036235                  EM2
001530  037256                  DH1             ; ERROR 2
001532  037434                  DT1
001534  036307                  EM3
001536  037317                  DH2             ; ERROR 3
001540  037456                  DT2
001542  036336                  EM4
001544  037256                  DH1             ; ERROR 4
001546  037434                  DT1
001550  036365                  EM5
001552  037256                  DH1             ; ERROR 5
001554  037434                  DT1
001556  036427                  EM6
001560  037334                  DH3             ; ERROR 6
001562  037434                  DT1
001564  036456                  EM7
001566  037334                  DH3             ; ERROR 7
001570  037434                  DT1
001572  036520                  EM10
001574  037334                  DH3             ; ERROR 10
001576  037434                  DT1
001600  036553                  EM11
001602  037317                  DH2             ; ERROR 11
001604  037456                  DT2
001606  036565                  EM12
001610  037317                  DH2             ; ERROR 12
001612  037456                  DT2
001614  036610                  EM13
001616  000000                  0               ; ERROR 13
001620  000000                  0
001622  036642                  EM14
001624  000000                  0               ; ERROR 14
001626  000000                  0
```

```
                                        EM15
                                        DH4             ; ERROR 15
                                        DT1
                                        EM16
                                        0               ; ERROR 16
                                        0
                                        EM17
                                        0               ; ERROR 17
                                        0
                                        EM20
                                        DH2             ; ERROR  20
                                        DT2
                                        EM21
                                        0               ; ERROR 21
                                        0
                                        EM22
                                        0               ; ERROR 22
                                        0
                                        EM23
                                        DH4             ; ERROR 23
                                        DT1
                                        EM24
                                        0               ; ERROR 24
                                        0
                                        EM25
                                        DH2             ; ERROR 25
                                        DT2
                    002034      .=2034
                                .SBTTL  APT PARAMETER BLOCK

                                ;;******************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;******************************************************************
            002034              .SX=.   ;;SAVE CURRENT LOCATION
            000024              .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024      000200              200     ;;FOR APT START UP
            000044              .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044      002034              $APTHDR ;;POINT TO APT HEADER BLOCK
            002034              .=.SX   ;;RESET LOCATION COUNTER
                                ;;******************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

002034                          $APTHD:
002034      000000              SHIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
002036      001316              SMBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
002040      000132              STSTM:  .WORD   90.     ;;RUN TIM OF LONGEST TEST
002042      000137              $PASTM: .WORD   95.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
002044      000137              $UNITM: .WORD   95.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
002046      000052              .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                ;KMC11 CONTROL INDICATORS FOR CURRENT KMC11 UNDER TEST
                                ;-----------------------------------------------------

002050  000000                  STAT1:  0
002052  000000                  STAT2:  0
002054  000000                  STAT3:  0

                                ;KMC11 VECTOR AND REGISTER INDIRECT POINTERS
                                ;-------------------------------------------

002056  000000                  KMRVEC: 0               ;POINTER TO KMC11 RECEIVER INTERRUPT VECTOR
002060  000000                  KMRLVL: 0               ;POINTER TO KMC11 RECEIVER INTERRUPT SERVICE PS
002062  000000                  KMTVEC: 0               ;POINTER TO KMC11 TRANSMITTER INTERRUPT VECTOR
002064  000000                  KMTLVL: 0               ;POINTER TO KMC11 TRANSMITTER INTERRUPT SERVICE PS
002066  000000                  KMCSR:  0               ;POINTER TO KMC11 CONTROL STATUS REGISTER
002070  000000                  KMCSRH: 0               ;POINTER TO KMC11 CONTROL STATUS REGISTER HIGH BYTE.
002072  000000                  KMCTL:  0               ;POINTER TO KMC11 CONTOL OUT REGISTER
002074  000000                  KMP04:  0               ;POINTER TO KMC11 PORT REGISTER(SEL 4)
002076  000000                  KMP06:  0               ;POINTER TO KMC11 PORT REGIS.ER(SEL 6)

                                ;TEMP STORAGE
                                ;------------

                                ;TEMP:  0
                                ;.=.+40

                                ;KMC11 STATUS TABLE AND ADDRESS ASSIGNMENTS
                                ;------------------------------------------

        002100                  .=2100
002100                          KM.MAP:
002100  000001                  KMCR00: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 00
002102  000001                  KMS100: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 00
002104  000001                  KMS200: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 00
002106  000001                  KMS300: .BLKW   1               ;3RD STATUS WORD

002110  000001                  KMCR01: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 01
002112  000001                  KMS101: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 01
002114  000001                  KMS201: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 01
002116  000001                  KMS301: .BLKW   1               ;3RD STATUS WORD

002120  000001                  KMCR02: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 02
002122  000001                  KMS102: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 02
002124  000001                  KMS202: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 02
002126  000001                  KMS302: .BLKW   1               ;3RD STATUS WORD

002130  000001                  KMCR03: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 03
002132  000001                  KMS103: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 03
002134  000001                  KMS203: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 03
002136  000001                  KMS303: .BLKW   1               ;3RD STATUS WORD

002140  000001                  KMCR04: .BLKW   1               ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 04
002142  000001                  KMS104: .BLKW   1               ;VECTOR FOR KMC11 NUMBER 04
002144  000001                  KMS204: .BLKW   1               ;DDCMP LINE# FOR KMC11 NUMBER 04
002146  000001                  KMS304: .BLKW   1               ;3RD STATUS WORD
```

```
                                
002150  000001          KMCR05: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 05
002152  000001          KMS105: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 05
002154  000001          KMS205: .BLKW   1       ;DDCMP LINE# FOR KMC11 NUMBER 05
002156  000001          KMS305: .BLKW   1       ;3RD STATUS WORD
                                
002160  000001          KMCR06: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 06
002162  000001          KMS106: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 06
002164  000001          KMS206: .BLKW   1       ;DDCMP LINE# FOR KMC11 NUMBER 06
002166  000001          KMS306: .BLKW   1       ;3RD STATUS WORD
                                
002170  000001          KMCR07: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 07
002172  000001          KMS107: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 07
002174  000001          KMS207: .BLKW   1       ;DDCMP LINE# FOR KMC11 NUMBER 07
002176  000001          KMS307: .BLKW   1       ;3RD STATUS WORD
                                
002200  000001          KMCR10: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 10
002202  000001          KMS110: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 10
002204  000001          KMS210: .BLKW   1       ;DDCMP LINE# FOR KMC11 NUMBER 10
002206  000001          KMS310: .BLKW   1       ;3RD STATUS WORD
                                
002210  000001          KMCR11: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 11
002212  000001          KMS111: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 11
002214  000001          KMS211: .BLKW   1       ;DDCMP LINE# FOR KMC11 NUMBER 11
002216  000001          KMS311: .BLKW   1       ;3RD STATUS WORD
                                
002220  000001          KMCR12: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 12
002222  000001          KMS112: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 12
002224  000001          KMS212: .BLKW   1       ;DDCMP LINE# FOR KMC11 NUMBER 12
002226  000001          KMS312: .BLKW   1       ;3RD STATUS WORD
                                
002230  000001          KMCR13: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 13
002232  000001          KMS113: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 13
002234  000001          KMS213: .BLKW   1       ;DDCMP LINE# FOR KMC11 NUMBER 13
002236  000001          KMS313: .BLKW   1       ;3RD STATUS WORD
                                
002240  000001          KMCR14: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 14
002242  000001          KMS114: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 14
002244  000001          KMS214: .BLKW   1       ;DDCMP LINE# FOR KMC11 NUMBER 14
002246  000001          KMS314: .BLKW   1       ;3RD STATUS WORD
                                
002250  000001          KMCR15: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 15
002252  000001          KMS115: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 15
002254  000001          KMS215: .BLKW   1       ;DDCMP LINE# FOR KMC11 NUMBER 15
002256  000001          KMS315: .BLKW   1       ;3RD STATUS WORD
                                
002260  000001          KMCR16: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 16
002262  000001          KMS116: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 16
002264  000001          KMS216: .BLKW   1       ;DDCMP LINE# FOR KMC11 NUMBER 16
002266  000001          KMS316: .BLKW   1       ;3RD STATUS WORD
                                
002270  000001          KMCR17: .BLKW   1       ;CONTROL STATUS REGISTER FOR KMC11 NUMBER 17
002272  000001          KMS117: .BLKW   1       ;VECTOR FOR KMC11 NUMBER 17
002274  000001          KMS217: .BLKW   1       ;DDCMP LINE# FOR KMC11 NUMBER 17
002276  000001          KMS317: .BLKW   1       ;3RD STATUS WORD
                                
```

```
  616  002300  000000                KM.END: 000000
```

```
                        ;KMC11 PASS COUNT AND ERROR COUNT TABLE
                        ;--------------------------------------

002302                  CNT.MAP:
002302  000000          PACT00: 0               ;PASS COUNT FOR KMC11 NUMBER 00
002304  000000          ERCT00: 0               ;ERROR COUNT FOR KMC11 NUMBER 00

002306  000000          PACT01: 0               ;PASS COUNT FOR KMC11 NUMBER 01
002310  000000          ERCT01: 0               ;ERROR COUNT FOR KMC11 NUMBER 01

002312  000000          PACT02: 0               ;PASS COUNT FOR KMC11 NUMBER 02
002314  000000          ERCT02: 0               ;ERROR COUNT FOR KMC11 NUMBER 02

002316  000000          PACT03: 0               ;PASS COUNT FOR KMC11 NUMBER 03
002320  000000          ERCT03: 0               ;ERROR COUNT FOR KMC11 NUMBER 03

002322  000000          PACT04: 0               ;PASS COUNT FOR KMC11 NUMBER 04
002324  000000          ERCT04: 0               ;ERROR COUNT FOR KMC11 NUMBER 04

002326  000000          PACT05: 0               ;PASS COUNT FOR KMC11 NUMBER 05
002330  000000          ERCT05: 0               ;ERROR COUNT FOR KMC11 NUMBER 05

002332  000000          PACT06: 0               ;PASS COUNT FOR KMC11 NUMBER 06
002334  000000          ERCT06: 0               ;ERROR COUNT FOR KMC11 NUMBER 06

002336  000000          PACT07: 0               ;PASS COUNT FOR KMC11 NUMBER 07
002340  000000          ERCT07: 0               ;ERROR COUNT FOR KMC11 NUMBER 07

002342  000000          PACT10: 0               ;PASS COUNT FOR KMC11 NUMBER 10
002344  000000          ERCT10: 0               ;ERROR COUNT FOR KMC11 NUMBER 10

002346  000000          PACT11: 0               ;PASS COUNT FOR KMC11 NUMBER 11
002350  000000          ERCT11: 0               ;ERROR COUNT FOR KMC11 NUMBER 11

002352  000000          PACT12: 0               ;PASS COUNT FOR KMC11 NUMBER 12
002354  000000          ERCT12: 0               ;ERROR COUNT FOR KMC11 NUMBER 12

002356  000000          PACT13: 0               ;PASS COUNT FOR KMC11 NUMBER 13
002360  000000          ERCT13: 0               ;ERROR COUNT FOR KMC11 NUMBER 13

002362  000000          PACT14: 0               ;PASS COUNT FOR KMC11 NUMBER 14
002364  000000          ERCT14: 0               ;ERROR COUNT FOR KMC11 NUMBER 14

002366  000000          PACT15: 0               ;PASS COUNT FOR KMC11 NUMBER 15
002370  000000          ERCT15: 0               ;ERROR COUNT FOR KMC11 NUMBER 15

002372  000000          PACT16: 0               ;PASS COUNT FOR KMC11 NUMBER 16
002374  000000          ERCT16: 0               ;ERROR COUNT FOR KMC11 NUMBER 16

002376  000000          PACT17: 0               ;PASS COUNT FOR KMC11 NUMBER 17
002400  000000          ERCT17: 0               ;ERROR COUNT FOR KMC11 NUMBER 17
```

```
670
671
672
673
674
675
```

FORMAT OF STATUS TABLE

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
C	O	N	T	R	O	L		R	E	G	I	S	T	E	R	CSR
*	*	*	*	*	*	*	*	*	V	E	C	T	O	R	*	STAT1
*	B	M			A	D	D	*	*	L	I	N	E		*	STAT2
															*	STAT3

DEFINITION OF FORMAT

CSR: CONTAINS KMC11 CSR ADDRESS

STAT1: BITS 00-08 IS KMC11 VECTOR ADDRESS
BIT14=1 ???? TURNAROUND CONNECTOR IS ON
BIT14=0 NO TURNAROUND CONNECTOR
BIT13=0 LINE UNIT IS AN M8201
BIT13=1 LINE UNIT IS AN M8202
BIT12=1 NO LINE UNIT
BITS 09-11 IS KMC11 BR PRIORITY LEVEL

STAT2: LOW BYTE IS SWITCH PAC#1 (DDCMP LINE NUMBER)
HIGH BYTE IS SWITCH PAC#2 (BM873 BOOT ADD)

STAT3: BIT0=1 DO FREE RUNNING TESTS ON KMC
(MUST BE SET TO A ONE MANUALLY [PROGRAMS G AND H ONLY])

```
                                        ;PROGRAM INITIALIZATION
                                        ;LOCK OUT INTERRUPTS
                                        ;SET UP PROCESSOR STACK
                                        ;SET UP POWER FAIL VECTOR
                                        ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
                                        ;TYPE TITLE MESSAGE

002402  012737  000340  177776  .START: MOV     #340,PS         ;LOCK OUT INTERRUPTS
002410  012706  001200          MOV     #STACK,SP       ;SET UP STACK
002414  012737  007100  000024  MOV     #$PWRDN,#24     ;SET UP POWER FAIL VECTOR
002422  012737  001472  001476  MOV     KWNUM,SAVNUM    ;SAVE NUMBER OF DEVICES IN SYSTEM.
002430  005067  011452          CLR     SWFLG           ;CLEAR SOFT TYPEOUT FLAG
002434  105067  001203          CLRB    ERRFLG          ;CLEAR ERROR FLAG
002440  105067  001511          CLRB    QV.FLG          ;ZERO QUICK VERIFY FLAG
002444  012737  002070  001502  MOV     #KM.MAP-10,CREAM;SET MAP POINTER.
002452  012737  002276  001504  MOV     #CNT.MAP-4,MILK ;SET PASS COUNT MAP POINTER
002460  012737  100000  001500  MOV     #BIT15,RUN      ;POINT POINTER TO FIRST DEVICE.
002466  012700  002302          MOV     #CNT.MAP,R0     ;PASS COUNT POINTER TO R0
002472  005020          23$:    CLR     (R0)+           ;CLEAR TABLE
002474  022700  002402          CMP     #CNT.MAP+100,R0 ;DONE YET?
002500  001374          BNE     23$             ;KEEP GOING
002502  005037  001216          CLR     SERRPC          ;CLEAR LAST ERROR POINTER
002506  012737  000001  001202  MOV     #1,STSTNM       ;SET UP FOR TEST 1
002514  012737  002402  001206  MOV     #.START,SLPADR  ;SET UP FOR POWER FAIL BEFORE
                                        ;TESTING STARTS
002522  132737  000001  001336  BITB    #1,SENV         ; IS IT RUNNING UNDER APT?
002530  001404          BEQ     3$              ; IF NOT CHECK FOR TYPE OF SWITCH REGISTER.
002532  013737  001340  000176  MOV     @SWREG,SWREG    ; LOAD SOFTWARE SWITCH REG.
002540  000423          BR      6$+2            ; GO SET UP SOFTWARE SWITCH REG.
002542  013746  000006          3$:     MOV     @#6,-(SP)       ;SAVE CURRENT VECTORS
002546  013746  000004          MOV     @#4,-(SP)       ;
002552  012737  002406  000004  MOV     #6$,@#4         ;SET UP FOR TIMEOUT
002560  012737  177570  001240  MOV     #177570,SWR     ;SET SWR TO HARD SWR ADDRESS
002566  012737  177570  001242  MOV     #177570,DISPLAY ;SET DISPLAY TO HARD SWR ADDRESS
002574  022777  177777  176436  CMP     #-1,@SWR        ;REFERENCE HARDWARE SWITCH REGISTER
002602  001402          BEQ     6$+2            ;IF = -1 USE SOFT SWR ANYWAY
002604  000407          BR      7$              ;IF IT EXISTS AND NOT = -1 USE HARD SWR
002606  022626          6$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
002610  012737  000176  001240  MOV     #SWREG,SWR      ;POINTER TO SOFT SWR
002616  012737  000174  001242  MOV     #DISPREG,DISPLAY;POINTER TO SOFT DISPLAY REG
002624  012637  000004          7$:     MOV     (SP)+,@#4       ;RESTORE VECTORS
002630  012637  000006          MOV     (SP)+,@#6       ;
002634  105737  001506          TSTB    INIFLG          ;HAS INITIALIZATION BEEN PERFORMED
002640  001006          BNE     20$             ;BR IF YES
002642  022737  004070  000042  CMP     #SENDAD,@#42    ;IF ACT-11 AUTOMATIC MODE, DON'T TYPE ID
002650  001402          BEQ     20$             ;
002652  104401  001000          TYPE    ,MTITLE         ;TYPE TITLE MESSAGE
002656  004737  011216  20$:     JSR     PC,CKSWR                ;CHECK FOR SOFT SWR
002662  017737  176352  001446  MOV     @SWR,STRTSW     ;STORE STARTING SWITCHES
002670  005737  000042          TST     @#42            ;IS IT RUNNING IN AUTO MODE?
002674  001402          BEQ     .+6             ;BR IF NO
002676  005037  001446          CLR     STRTSW          ;IF YES, CLEAR SWITCHES
002702  032737  000001  001446  BIT     #SW00,STRTSW    ;IF SW00=1, QUESTIONS ARE ASKED.
002710  001012          BNE     17$             ;BR IF SW00=1
002712  105737  001446          TSTB    STRTSW          ;BIT7=1??
```

```
 780  002716  100007                        BPL     17$             ;BR IF SW07=0
 781  002720  005737  001470                TST     KMACTV          ;ARE ANY DEVICES SELECTED?
 782  002724  001027                        BNE     16$             ;BR IF YES
 783  002726  104401  010703                TYPE,   NOACT           ;NO DEVICES SELECTED.
 784  002732  000000                        HALT                    ;STOP THE SHOW
 785  002734  000776                        BR      .-2             ;DISQUALIFY CONTINUE SWITCH
 786  002736  105737  001336        17$:    TSTB    $ENV            ;  IS IT UNDER APT DUMP MODE?
 787  002742  001405                        BEQ     27$             ;  YES, CHECK IF APT SIZED IT?
 788  002744  132737  000001  001336        BITB    #1,$ENV         ;  IS IT UNDER ODT, V OR RUN MODE?
 789  002752  001012                        BNE     30$             ;  YES, NEEDS ONLY APT SIZING.
 790  002754  000405                        BR      33$             ;  NO, NEEDS REGULAR AUTO.SIZE.
 791  002756  105737  001337        27$:    TSTB    $ENVM           ;  IS IT SIZED BY APT?
 792  002762  100406                        BMI     30$             ;  YES, NEEDS ONLY APT SIZING.
 793  002764  042737  000001  001446        BIC     #SWDO.STRTSW    ;  SIZE ONLY IN AUTO MODE.
 794  002772  004737  012152        33$:    JSR     PC,AUTO.SIZE    ;  GO DO THE AUTO.SIZE.
 795  002776  000402                        BR      16$             ;  GO PRINT THE MAP.
 796  003000  004737  013552        30$:    JSR     PC,APT.SIZE     ;  GO DO THE APT SIZING.
 797  003004  105737  001506        16$:    TSTB    INIFLG          ;FIRST TIME?
 798  003010  001410                        BEQ     21$             ;BR IF YES
 799  003012  105737  001446                TSTB    STRTSW          ;IF USING SAME PARAMETERS DONT TYPE MAP
 800  003016  100431                        BMI     1$
 801  003020  032737  000006  001446        BIT     #BIT1!BIT2,STRTSW;IS TEST NO. OR LOCK SELECTED
 802  003026  001403                        BEQ     24$             ;IF NO THEN TYPE STATUS
 803  003030  000424                        BR      1$              ;IF YES DO NOT TYPE STATUS
 804  003032  105137  001506        21$:    COMB    INIFLG          ;SET FLAG
 805  003036  104401  010051        24$:    TYPE    ,XHEAD          ;TYPE HEADER
 806  003042  012704  002100                MOV     #KM.MAP,R4      ;SET POINTER
 807  003046  010437  001276        5$:     MOV     R4,STMP0        ;SET ADDRESS
 808  003052  012437  001300                MOV     (R4)+,STMP1     ;SET CSR
 809  003056  001411                        BEQ     1$              ;ALL DONE IF ZERO
 810  003060  012437  001302                MOV     (R4)+,STMP2     ;SET STAT1
 811  003064  012437  001304                MOV     (R4)+,STMP3     ;SET STAT2
 812  003070  012437  001306                MOV     (R4)+,STMP4     ;SET STAT3
 813  003074  104416                        CONVRT                  ;TYPE OUT STATUS MAP
 814  003076  011064                        XSTAT@                  ;
 815  003100  000762                        BR      5$
 816  003102  012700  002100        1$:     MOV     #KM.MAP,R0      ;R0 POINTS TO STATUS TABLE
 817
 818                                ;;#################################################################################
 819                                ;;#AUTO SIZE TEST
 820                                ;;#THIS TEST VERIFYS THAT THE KMC11S AND/OR KMC11S ARE AT THE CORRECT FLOATING
 821                                ;;#ADDRESSES FOR YOUR SYSTEM. IF THIS TEST FAILS, IT IS NOT A HARDWARE ERROR.
 822                                ;;#CHECK THE ADDRESSES OF ALL FLOATING DEVICES (DJ,DH,DQ,DU,DUP,LK,DMC,DZ,KMC).
 823                                ;;#IF THERE ARE NO OTHER FLOATING DEVICES BEFORE THE KMC11, THE FIRST
 824                                ;;#KMC11 IS 760110. NO DEVICE SHOULD EVER BE AT
 825                                ;;#ADDRESS 760000.
 826                                ;;#################################################################################
 827
 828  003106  013746  000004                MOV     @#4,-(SP)       ;SAVE LOC 4
 829  003112  013746  000006                MOV     @#6,-(SP)       ;SAVE LOC 6
 830  003116  005037  000006                CLR     @#6             ;CLEAR VEC+2
 831  003122  005037  001302                CLR     STMP2           ;CLEAR FLAG
 832  003126  011037  002066        AUSTRT: MOV     (R0),KMCSR      ;GET NEXT KMC CSR
 833  003132  001510                        BEQ     AUDONE          ;BR IF DONE
 834  003134  012737  003240  000004  2$:    MOV     #NODEV,@#4      ;SET UP FOR TIMEOUT
 835  003142  012703  000010        3$:     MOV     #10,R3          ;R3 IS COUNT OF DEVICES BEFORE KMC
```

```
                                                
```

```
                003354                  .EVEN
003354                          AUDONE:
003354  012637  000006          1$:     MOV     (SP)+,@#6       ;RESTORE LOC 6
003360  012637  000004                  MOV     (SP)+,@#4       ;RESTORE LOC 4
003364  032737  000010  001446          BIT     #SW03,STRTSW    ;SELECT SPECIFIC DEVICES??
003372  001422                          BEQ     3$              ;BR IF NO.
003374  104401  007771                  TYPE    ,MNEW           ;TYPE THE MESSAGE.
003400  005000                          CLR     R0              ;ZERO DATA LIGHTS
003402  000000                          HALT                    ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
003404  027737  175630  001474          CMP     @SWR,SAVACT     ;IS THE NUMBER VALID?
003412  101404                          BLOS    2$              ;BR IF NUMBER IS OK.
003414  104401  007644                  TYPE    ,MERR3          ;TELL USER OF INVALID NUMBER.
003420  000000                          HALT                    ;STOP EVERY THING.
003422  000776                          BR      .-2             ;RESTART THE PROGRAM AGAIN.
003424  017737  175610  001470  2$:     MOV     @SWR,KMACTV     ;GET NEW DEVICE PATTERN
003432  013700  001470                  MOV     KMACTV,R0       ;SHOW THE USER WHAT HE SELECTED.
003436  000000                          HALT                    ;CONTINUE DYNAMIC SWITCHES.
003440  012700  000300          3$:     MOV     #300,R0         ;PREPARE TO CLEAR THE FLOATING
003444  012701  000302                  MOV     #302,R1         ;VECTOR AREA. 300-776
003450  010120                  4$:     MOV     R1,(R0)+        ;START PUTTING "PC+2 - HALT"
003452  005021                          CLR     (R1)+           ;IN VECTOR AREA.
003454  022021                          CMP     (R0)+,(R1)+     ;POP POINTERS
003456  022700  001000                  CMP     #1000,R0        ;ALL DONE??
003462  001372                          BNE     4$              ;BR IF NO.

                                ;TEST START AND RESTART
                                ;----------------------

003464  012706  001200          .BEGIN: MOV     #STACK,SP       ;SET UP STACK
003470  013746  000006                  MOV     @#6,-(SP)       ;SAVE LOC 6
003474  013746  000004                  MOV     @#4,-(SP)       ;SAVE LOC 4
003500  005000                          CLR     R0              ;START AT 0
003502  012737  003546  000004          MOV     #2$,@#4         ;SET UP FOR TIME OUT
003510  005037  000006                  CLR     @#6             ;TO AUTOSIZE MEMORY
003514  005720                  6$:     TST     (R0)+           ;CHECK ADDRESS IN R0
003516  022700  157776                  CMP     #157776,R0      ;IS IT AT LEAST 28K
003522  001374                          BNE     6$              ;BR IF NO
003524  162700  007776                  SUB     #7776,R0        ;SAVE 2K FOR MONITORS
003530  010037  001466          7$:     MOV     R0,MEMLIM       ;STORE MEMORY LIMIT
003534  012637  000004                  MOV     (SP)+,@#4       ;RESTORE LOC 4
003540  012637  000006                  MOV     (SP)+,@#6       ;RESTORE LOC 6
003544  000413                          BR      10$             ;CONTINUE
003546  022626                  2$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
003550  162700  000004                  SUB     #4,R0           ;GET LAST GOOD ADDRESS
003554  162700  007776                  SUB     #7776,R0        ;SAVE 2K FOR MONITORS
003560  022700  030000                  CMP     #30000,R0       ;IS IT 8K?
003564  001361                          BNE     7$              ;BR IF NO
003566  012700  037400                  MOV     #37400,R0       ;IF 8K DON'T SAVE 2K
003572  000756                          BR      7$              ;
003574  012737  000340  177776  10$:     MOV     #340,PS         ;LOCK OUT INTERRUPTS
003602  032737  000004  001446          BIT     #BIT2,STRTSW    ;CHECK FOR LOCK ON TEST
003610  001406                          BEQ     1$              ;BR IF NO LOCK DESIRED.
003612  104401  007670                  TYPE    ,MLOCK          ;TYPE LOCK SELECTED.
003616  012737  000240  004146          MOV     #NOP,TTST       ;SET UP TO LOCK
003624  000403                          BR      3$              ;CONTINUE ALONG.
003626  013737  004354  004146  1$:     MOV     BRW,TTST        ;PREPARE NORMAL SCOPE ROUTINE
```

```
 948  003634  012737  011464  001206  3$:     MOV     #CYCLE,SLPADR   ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
 949  003642  032737  000002  001446  4$:     BIT     #SW01,STRTSW    ;IS TEST NO. SELECTED?
 950  003650  001002                          BNE     5$              ;BR IF YES
 951  003652  104401  007614                  TYPE    ,MR             ;TYPE R
 952  003656  000177  175324          5$:     JMP     @SLPADR         ;START TESTING
```

```
                                ;END OF PASS
                                ;TYPE NAME OF TEST
                                ;UPDATE PASS COUNT
                                ;CHECK FOR EXIT TO ACT-11
                                ;RESTART TEST

                        .SBTTL  END OF PASS ROUTINE

                        ;;**************************************************************************
                        ;*INCREMENT THE PASS NUMBER ($PASS)
                        ;*IF THERES A MONITOR GO TO IT
                        ;*IF THERE ISN'T JUMP TO CYCLE

003662                  $EOP:
                                RESET
003662  000005
003664  005237  001324          INC     $PASS           ;INCREMENT THE PASS COUNT
003670  105037  001203          CLRB    SERFLG          ;CLEAR ERROR FLAG
003674  104401  007572          TYPE    ,MEPASS         ;TYPE END PASS.
003700  104401  007717          TYPE    ,MCSRX          ;TYPE "CSR"
003704  104417  004104          CNVRT   ,XCSR           ;SHOW IT.
003710  104401  007725          TYPE    ,MVECX          ;TYPE VECTOR.
003714  104417  004112          CNVRT   ,XVEC           ;SHOW IT.
003720  104401  007733          TYPE    ,MPASSX         ;TYPE " PASSES "
003724  104417  004120          CNVRT   ,XPASS          ;SHOW IT.
003730  104401  007744          TYPE    ,MERRX          ;TYPE " ERRORS "
003734  104417  004126          CNVRT   ,XERR           ;SHOW IT.
003740  013700  001504          MOV     #ILK,R0         ;SET POINTER TO PASSCNT.
003744  013720  001324          MOV     $PASS,(R0)+     ;SAVE THE PASS COUNT.
003750  013720  001212          MOV     SERTTL,(R0)+    ;SAVE ERROR COUNT
003754  013777  002060  176074  MOV     @KWRLVL,@#KWRVEC ;RESTORE THE RECEIVER INTERRUPT VECTOR.
003762  005077  176072          CLR     @KWRLVL         ;RESTORE RECEIVER LEVEL
003766  013777  002064  176066  MOV     @KWTLVL,@#KWTVEC ;RESTORE THE TRANSMIT INTERRUPT VECTOR.
003774  005077  176064          CLR     @KWTLVL         ;RESTORE TRANSMITTER LEVEL
004000  005337  001476          DEC     SAVNUM          ;ALL DEVICE TESTED?
004004  001035                  BNE     $DOAGN          ;BRANCH IF NO.
004006  112737  000377  001511  MOVB    #377,QV.FLG     ;SET QUICK VERIFY FLAG.
004014  013737  001472  001476  MOV     KWNUM,SAVNUM    ;RESTORE DEVICE COUNT.
004022  005037  001216          CLR     SERRPC          ;CLEAR LAST ERROR PC
004026  005037  001310          CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
004032  005237  001324          INC     $PASS           ;;INCREMENT THE PASS NUMBER
004036  042737  100000  001324  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
004044  005327                  DEC     (PC)+           ;;LOOP?
004046  000001          $EOPCT: .WORD   1
004050  003013                  BGT     $DOAGN          ;;YES
004052  012737                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
004054  000001          $ENDCT: .WORD   1
004056  004046                  $EOPCT
004060  013700  000042  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
004064  001405                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
004066  000005                  RESET                   ;;CLEAR THE WORLD
004070  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
004072  000240                  NOP                     ;;SAVE ROOM
004074  000240                  NOP                     ;;FOR
004076  000240                  NOP                     ;;ACT11
004100                  $DOAGN:
004100  000137                  JMP     @(PC)+          ;;RETURN
```

```
                                ;SRTNAD: .WORD   CYCLE
                                ;XCSR:   1
                                ;        .BYTE   6,2
                                ;        KMCSR
                                ;XVEC:   1
                                ;        .BYTE   4,2
                                ;        KMRVEC
                                ;XPASS:  1
                                ;        .BYTE   6,2
                                ;        $PASS
                                ;XERR:   1
                                ;        .BYTE   6,2
                                ;        $ERTTL

                                ;SCOPE LOOP AND INTERATION HANDLER
                                ;--------------------------------

                                .SBTTL  SCOPE HANDLER ROUTINE

                                ;;*****************************************************************
                                ;#THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                ;#AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                ;#AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                ;#THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;#SW14=1         LOOP ON TEST
                                ;#SW11=1         INHIBIT ITERATIONS
                                ;#CALL
                                ;#      SCOPE           ;;SCOPE=IOT

004134                          $SCOPE:
004134  005037  001216          TTST:   CLR     $ERRPC          ; CLEAR LAST ERROR PC
004140  023716  013776                  CMP     TST1+2,(SP)     ; IS THIS TEST #1 ?
004144  001413                          BEQ     $XTSTR          ; IF SO DON'T LOOP.
004146  000406                          BR      1$              ;
004150  105777  175070                  TSTB    @$TKS           ; KEYBOARD DONE ?
004154  100067                          BPL     $OVER           ; IF NO DONT WAIT.
004156  017766  175064  177776          MOV     @$TKB,-2(SP)    ;
004164  032777  040000  175046  1$:     BIT     #BIT14,@$SWR    ;;LOOP ON PRESENT TEST?
004172  001060                          BNE     $OVER           ;;YES IF SW14=1
                                ;#####START OF CODE FOR THE XOR TESTER#####
004174  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
004176  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
004202  012737  004222  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
004210  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
004214  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
004220  000436                          BR      $SVLAD          ;;GO TO THE NEXT TEST
004222  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
004224  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
004230  000441                          BR      $OVER           ;;LOOP ON THE PRESENT TEST
004232                          6$:;#####END OF CODE FOR THE XOR TESTER#####
004232  105737  001203          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
004236  001404                          BEQ     3$              ;;BR IF NO
004240  105037  001203          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
004244  005037  001310                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
004250  032777  004000  174762  3$:     BIT     #BIT11,@$SWR    ;;INHIBIT ITERATIONS?
```

```
                                                BNE     1$              ;;BR IF YES
                                                TST     $PASS           ;;IF FIRST PASS OF PROGRAM
                                                BEQ     1$              ;;       INHIBIT ITERATIONS
                                                INC     $ICNT           ;;INCREMENT ITERATION COUNT
                                                CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
                                                BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
                                        1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
                                                MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
                                        $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
                                                MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
                                                MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
                                        $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
                                                MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
                                                MOV     KMCSR,R1                ; R1 CONTAINS BASE KMC ADDRESS.
                                                RTI
                                        BRW:    .WORD   406
                                        $MXCNT: 20                      ;;MAX. NUMBER OF ITERATIONS

                                                ;CHECK FOR FREEZE ON CURRENT DATA
                                                ;--------------------------------

                                        .SCOP1: JSR     PC,CKSWR                ;CHECK FOR SOFT SWR
                                                BIT     #SW09,@SWR      ;IS SW09=1(SET)?
                                                BEQ     1$              ;BR IF NOT SET.
                                                MOV     #-1,LOCK        ;FLAG IT.
                                        1$:     RTI                     ;GO BACK.

                                                ;TELETYPE OUTPUT ROUTINE
                                                ;-----------------------

                                                .SBTTL  TYPE ROUTINE

                                                ;;**************************************************************
                                                ;#ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                                ;#THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                                ;#NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                                ;#NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                                ;#NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                                ;#
                                                ;#CALL:
                                                ;#1) USING A TRAP INSTRUCTION
                                                ;#       TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                                ;#OR
                                                ;#       TYPE
                                                ;#       MESADR
                                                ;#

                                        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
                                                BPL     1$              ;;BR IF YES
                                                HALT                    ;;HALT HERE IF NO TERMINAL
                                                BR      3$              ;;LEAVE
                                        1$:     MOV     R0,-(SP)        ;;SAVE R0
                                                MOV     22(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
                                                CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
                                                BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
                                                BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
```

```
1121  004442  001405                    BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
1122  004444  010037  004454            MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
1123  004450  004737  004674            JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
1124  004454  000000            61$:    .WORD   0               ;;MESSAGE ADDRESS
1125  004456  132737  000040  001337    BITB    #APTCSUP,SENVM  ;;APT CONSOLE SUPPRESSED
1126  004464  001003                    BNE     60$             ;;YES,SKIP TYPE OUT
1127  004466  112046            2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
1128  004470  001005                    BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
1129  004472  005726                    TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
1130  004474  012600            60$:    MOV     (SP)+,R0        ;;RESTORE R0
1131  004476  062716  000002    3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
1132  004502  000002                    RTI                     ;;RETURN
1133  004504  122716  000011    4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
1134  004510  001430                    BEQ     8$
1135  004512  122716  000200            CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
1136  004516  001006                    BNE     5$
1137  004520  005726                    TST     (SP)+           ;;POP  <CR><LF> EQUIV
1138  004522  104401                    TYPE                    ;;TYPE A CR AND LF
1139  004524  001313                    .CRLF
1140  004526  105037  004662            CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
1141  004532  000755                    BR      2$              ;;GET NEXT CHARACTER
1142  004534  004737  004616    5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
1143  004540  123726  001256    6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
1144  004544  001350                    BNE     2$              ;;IF NO GO GET NEXT CHAR.
1145  004546  013746  001254            MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
1146                                                            ;;AND THE NULL CHAR.
1147  004552  105366  000001    7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
1148  004556  002770                    BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
1149  004560  004737  004616            JSR     PC,$TYPEC       ;;GO TYPE A NULL
1150  004564  105337  004662            DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
1151  004570  000770                    BR      7$              ;;LOOP
1152
1153                                    ;HORIZONTAL TAB PROCESSOR
1154
1155  004572  112716  000040    8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
1156  004576  004737  004616    9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
1157  004602  132737  000007  004662    BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
1158  004610  001372                    BNE     9$              ;;TAB STOP
1159  004612  005726                    TST     (SP)+           ;;POP SPACE OFF STACK
1160  004614  000751                    BR      2$              ;;GET NEXT CHARACTER
1161  004616  105777  174426    $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
1162  004622  100375                    BPL     $TYPEC
1163  004624  116677  000002  174420    MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
1164  004632  122766  000015  000002    CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
1165  004640  001003                    BNE     1$              ;;BRANCH IF NO
1166  004642  105037  004662            CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
1167  004646  000406                    BR      $TYPEX          ;;EXIT
1168  004650  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
1169  004656  001402                    BEQ     $TYPEX          ;;BRANCH IF YES
1170  004660  105227                    INCB    (PC)+           ;;COUNT THE CHARACTER
1171  004662  000000            $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
1172  004664  000207            $TYPEX: RTS     PC
1173
1174                                    .SBTTL  APT COMMUNICATIONS ROUTINE
1175
1176                                    ;;**************************************************************************
```

```
1177  004666  112737  000001  005132  $ATY1:  MOVB    #1,$EFLG        ;;TO REPORT FATAL ERROR
1178  004674  112737  000001  005130  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
1179  004702  000403                  BR      $ATYC
1180  004704  112737  000001  005132  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
1181  004712                          $ATYC:
1182  004712  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
1183  004714  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
1184  004716  105737  005130          TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
1185  004722  001450                  BEQ     5$              ;;IF NOT:  BR
1186  004724  122737  000001  001336  CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
1187  004732  001031                  BNE     3$              ;;IF NOT:  BR
1188  004734  132737  000100  001337  BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
1189  004742  001425                  BEQ     2$              ;;IF NOT:  BR
1190  004744  017600  000004          MOV     4(SP),R0        ;;GET MESSAGE ADDR.
1191  004750  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
1192  004756  005737  001316  1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
1193  004762  001375                  BNE     1$              ;;IF NOT:  WAIT
1194  004764  010037  001332          MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
1195  004770  105720          2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
1196  004772  001376                  BNE     2$
1197  004774  163700  001332          SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
1198  005000  006200                  ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
1199  005002  010037  001334          MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
1200  005006  012737  000004  001316  MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
1201  005014  000413                  BR      5$
1202  005016  017637  000004  005042  3$:     MOV     4(SP),4$        ;;PUT MSG ADDR IN JSR LINKAGE
1203  005024  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
1204  005032  013746  177776          MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
1205  005036  004737  004404          JSR     PC,$TYPE        ;;CALL TYPE MACRO
1206  005042  000000          4$:     .WORD   0
1207  005044                  5$:
1208  005044  105737  005132  10$:     TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
1209  005050  001416                  BEQ     12$             ;;IF NOT:  BR
1210  005052  005737  001336          TST     $ENV            ;;RUNNING UNDER APT?
1211  005056  001413                  BEQ     12$             ;;IF NOT:  BR
1212  005060  005737  001316          TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
1213  005064  001375                  BNE     11$             ;;IF NOT:  WAIT
1214  005066  017637  000004  001320  MOV     4(SP),$FATAL    ;;GET ERROR #
1215  005074  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
1216  005102  005237  001316          INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
1217  005106  105037  005132  12$:     CLRB    $FFLG           ;;CLEAR FATAL FLAG
1218  005112  105037  005131          CLRB    $LFLG           ;;CLEAR LOG FLAG
1219  005116  105037  005130          CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
1220  005122  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
1221  005124  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
1222  005130  000207                  RTS     PC              ;;RETURN
1223  005132  000     $MFLG:  .BYTE   0               ;;MESSG. FLAG
1224  005131  000     $LFLG:  .BYTE   0               ;;LOG FLAG
1225  005132  000     $FFLG:  .BYTE   0               ;;FATAL FLAG
1226  005134                  .EVEN
1227          000200  APTSIZE=200
1228          000001  APTENV=001
1229          000100  APTSPOOL=100
1230          000040  APTCSUP=040
1231                  ;-------------------------
1232
```

```
                                .SBTTL  TTY INPUT ROUTINE
                                ;;*****************************************************************
                                .ENABL  LSB

                                .DSABL  LSB


                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                ;
005134  011646                  SRDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
005136  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
005144  105777  174074          1$:     TSTB    @#TKS           ;;WAIT FOR
005150  100375                          BPL     1$              ;;A CHARACTER
005152  117766  174070  000004          MOVB    @#TKB,4(SP)     ;;READ THE TTY
005160  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
005166  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
005174  001013                          BNE     3$              ;;BRANCH IF NO
005176  105777  174042          2$:     TSTB    @#TKS           ;;WAIT FOR A CHARACTER
005202  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
005204  117746  174036                  MOVB    @#TKB,-(SP)     ;;GET CHARACTER
005210  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
005214  026627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
005220  001366                          BNE     2$              ;;IF NOT DISCARD IT
005222  000750                          BR      1$              ;;YES, RESUME
005224  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
005232  002407                          BLT     4$              ;;BRANCH IF YES
005234  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
005242  003003                          BGT     4$              ;;BRANCH IF YES
005244  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
005252  000002                  4$:     RTI                     ;;GO BACK TO USER
                                ;;*****************************************************************
                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                ;*CALL:
                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

005254  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
005256  005046                          CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
005260  012703  005510          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
005264  022703  005517          2$:     CMP     #$TTYIN+7,R3    ;;BUFFER FULL?
005270  101456                          BLOS    4$              ;;BR IF YES
005272  104403                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
005274  112313                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
005276  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
005302  001022                          BNE     5$              ;;BR IF NO
005304  005716                          TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
005306  001007                          BNE     6$              ;;BR IF NO
005310  112737  000134  005506          MOVB    #'\,9$          ;;TYPE A BACK SLASH
```

```
1289  005316  104401  005506          TYPE    9$
1290  005322  012716  177777          MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
1291  005326  005303          6$:     DEC     R3              ;;BACKUP BY ONE
1292  005330  021327  005510          CMP     R3,#STTYIN      ;;STACK EMPTY?
1293  005334  103434                  BLO     4$              ;;BR IF YES
1294  005336  111337  005506          MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
1295  005342  104401  005506          TYPE    9$              ;;GO TYPE
1296  005346  000746                  BR      2$              ;;GO READ ANOTHER CHAR.
1297  005350  005716          5$:     TST     (SP)            ;;RUBOUT KEY SET?
1298  005352  001406                  BEQ     7$              ;;BR IF NO
1299  005354  112737  000134  005506  MOVB    #'\,9$          ;;TYPE A BACK SLASH
1300  005362  104401  005506          TYPE    9$
1301  005366  005016                  CLR     (SP)            ;;CLEAR THE RUBOUT KEY
1302  005370  122713  000025  7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
1303  005374  001003                  BNE     8$              ;;BR IF NO
1304  005376  104401  005517          TYPE    $CNTLU          ;;TYPE A CONTROL "U"
1305  005402  000726                  BR      1$              ;;GO START OVER
1306  005404  122713  000022  8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
1307  005410  001011                  BNE     3$              ;;BRANCH IF NO
1308  005412  105013                  CLRB    (R3)            ;;CLEAR THE CHARACTER
1309  005414  104401  001313          TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
1310  005420  104401  005510          TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
1311  005424  000717                  BR      2$              ;;GO PICKUP ANOTHER CHACTER
1312  005426  104401  001312  4$:     TYPE    $QUES           ;;TYPE A '?'
1313  005432  000712                  BR      1$              ;;CLEAR THE BUFFER AND LOOP
1314  005434  111337  005506  3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
1315  005440  104401  005506          TYPE    9$
1316  005444  122723  000015          CMPB    #15,(R3)+       ;;CHECK FOR RETURN
1317  005450  001305                  BNE     2$              ;;LOOP IF NOT RETURN
1318  005452  105063  177777          CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
1319  005456  104401  001314          TYPE    $LF             ;;TYPE A LINE FEED
1320  005462  005726                  TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
1321  005464  012603                  MOV     (SP)+,R3        ;;RESTORE R3
1322  005466  011646                  MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
1323  005470  016666  000004  000002  MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
1324  005476  012766  005510  000004  MOV     #STTYIN,4(SP)
1325  005504  000002                  RTI                     ;;RETURN
1326  005506  000             9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
1327  005507  000             $TERMB: .BYTE   0               ;;TERMINATOR
1328  005510  000007          STTYIN: .BLKB   7               ;;RESERVE 7 BYTES FOR TTY INPUT
1329  005517  136  025  005525  000012  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
1330  005524  043536  005015  000  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
1331  005531  015  051412  051127  $NSWR:  .ASCIZ  <15><12>/SWR = /
1332  005536  036440  000040
1333  005542  020040  042516  020127  $NNEW:  .ASCIZ  /  NEW = /
1334  005550  020075  000
1335  005554
1336                          .EVEN
1337                          .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
1338
1339                          ;;*****************************************************************
1340                          ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
1341                          ;*CHANGE IT TO BINARY.
1342                          ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
1343                          ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
1344                          ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
                              ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
```

```
                        ;*CALL:
                        ;*      RDOCT                   ;;READ AN OCTAL NUMBER
                        ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                        ;*                              ;;HIGH ORDER BITS ARE IN SHIOCT
                        
005554  011646          SRDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
005556  016666  000004  000002        MOV     4(SP),2(SP)     ;;INPUT NUMBER
005564  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
005566  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
005570  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
005572  104403                1$:     RDLIN                   ;;READ AN ASCIZ LINE
005574  012600                        MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
005576  010037  005702                MOV     R0,5$           ;;AND SAVE IT
005602  005001                        CLR     R1              ;;CLEAR DATA WORD
005604  005002                        CLR     R2
005606  112046                2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
005610  001420                        BEQ     3$              ;;IF ZERO GET OUT
005612  122716  000060                CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
005616  003026                        BGT     4$              ;;IS AN OCTAL DIGIT
005620  122716  000067                CMPB    #'7,(SP)
005624  002423                        BLT     4$
005626  006301                        ASL     R1              ;;*2
005630  006102                        ROL     R2
005632  006301                        ASL     R1              ;;*4
005634  006102                        ROL     R2
005636  006301                        ASL     R1              ;;*8
005640  006102                        ROL     R2
005642  042716  177770                BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
005646  062601                        ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
005650  000756                        BR      2$              ;;LOOP
005652  005726                3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
005654  010166  000012                MOV     R1,12(SP)       ;;SAVE THE RESULT
005660  010267  005712                MOV     R2,$HIOCT
005664  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
005666  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
005670  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
005672  000002                        RTI                     ;;RETURN
005674  005726                4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
005676  105010                        CLRB    (R0)            ;;SET A TERMINATOR
005700  104401                        TYPE                    ;;TYPE UP THRU THE BAD CHAR.
005702  000000                5$:     .WORD   0
005704  104401  001312                TYPE    $QUES           ;;"?" "CR" & "LF"
005710  000730                        BR      1$              ;;TRY AGAIN
005712  000000                $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
                        ;
                        ;       INPUT OCTAL NUMBER ROUTINE
                        ;-------------------------------
                        ;
005714  010546          $INPUT: MOV     R5,-(SP)        ; SAVE REGISTER R5.
005716  016605  000002        MOV     2($P),R5        ; GET FIRST PARAMETER ADDRESS.
005722  012537  005760        MOV     (R5)+,WHAT      ; GET MESSAGE ADDRESS.
005726  012537  006040        MOV     (R5)+,LOLIM     ; GET LOW LIMIT FOR THE #
005732  012537  006042        MOV     (R5)+,HILIM     ; GET HIGH LIMIT FOR THE #.
005736  012537  006044        MOV     (R5)+,WHERE     ; GET ADDRESS OF INBUFFER
005742  112537  006046        MOVB    (R5)+,LOBITS    ; GET LOWMASK BITS,
005746  112537  006047        MOVB    (R5)+,ADRCNT    ; GET # OF #'S TO BE GENERATED.
```

```
1401  005752  010566  000002          MOV     R5,2(SP)                ;SAVE THE RETURN ADDRESS.
1402  005756  104401                  INLP1:  TYPE                    ;TYPE THE MESSAGE.
1403  005760  000000                  WHAT:   .WORD   0
1404  005762  104404                          RDOCT                   ;READ OCTAL # FROM KEYBOARD.
1405  005764  021637  006042                  CMP     (SP),HILIM      ;IS IT IN HIGH LIMIT?
1406  005770  003003                          BGT     2$              ;BRANCH IF NO.
1407  005772  021637  006040                  CMP     (SP),LOLIM      ;IS IT MORE THAN LOW LIMIT.
1408  005776  002005                          BGE     3$              ;BRANCH IF YES.
1409  006000  104401  001312          2$:     TYPE    ,$QUES          ;TYPE "?"
1410  006004  104401  001313                  TYPE    ,$CRLF          ;TYPE <CR>,<LF>
1411  006010  000762                          BR      INLP1
1412  006012  013705  006044          3$:     MOV     WHERE,R5        ;GET BUFFER ADDRESS.
1413  006016  011625                  4$:     MOV     (SP),(R5)+      ;SAVE THE # IN RIGHT PLACE.
1414  006020  062716  000002                  ADD     #2,(SP)         ;NEXT SEQUENTIAL NUMBER.
1415  006024  105337  006047                  DECB    ADRCNT          ;COUNT BY 1.
1416  006030  001372                          BNE     4$              ;BRANCH IF NOT DONE.
1417  006032  005726                          TST     (SP)+           ;POP THE STACK POINTER.
1418  006034  012605                          MOV     (SP)+,R5        ;POP THE REG.5
1419  006036  000002                          RTI
1420  006040  000000                  LOLIM:  .WORD   0
1421  006042  000000                  HILIM:  .WORD   0
1422  006044  000000                  WHERE:  .WORD   0
1423  006046     000                  LOBITS: .BYTE   0
1424  006047     000                  ADRCNT: .BYTE   0
1425
1426                                  ;  ADVANCE TO NEXT TEST HANDLER
1427                                  ;-------------------------------
1428                                  ;
1429  006050  013716  001442  .ADVANCE:        MOV     NEXT,(SP)               ; CRUNCH STACK WITH ADDRESSOF SCOPE CALL
1430  006054  005037  001444          CLR     LOCK                    ; RESET TIGHT LOOP ADDRESS
1431  006060  000002                  RTI                             ; CHECK TO SEE IF OLD TEST GETS REPEATED
1432
1433                                  ;SAVE PC OF TEST THAT FAILED AND R0-R5
1434                                  ;-------------------------------------
1435
1436  006062  016637  000004  001460  .SAV05: MOV     4(SP),SAVPC     ;SAVE R7 (PC)
1437
1438                                  ;SAVE R0-R5
1439
1440  006070  010537  001274          SV05:   MOV     R5,SREG5        ;SAVE R5
1441  006074  010437  001272                  MOV     R4,SREG4        ;SAVE R4
1442  006100  010337  001270                  MOV     R3,SREG3        ;SAVE R3
1443  006104  010237  001266                  MOV     R2,SREG2        ;SAVE R2
1444  006110  010137  001264                  MOV     R1,SREG1        ;SAVE R1
1445  006114  010037  001262                  MOV     R0,SREG0        ;SAVE R0
1446  006120  000002                          RTI                     ;LEAVE.
1447
1448                                  ;RESTORE R0-R5
1449
1450  006122  013700  001262          .RES05: MOV     SREG0,R0        ;RESTORE R0
1451  006126  013701  001264                  MOV     SREG1,R1        ;RESTORE R1
1452  006132  013702  001266                  MOV     SREG2,R2        ;RESTORE R2
1453  006136  013703  001270                  MOV     SREG3,R3        ;RESTORE R3
1454  006142  013704  001272                  MOV     SREG4,R4        ;RESTORE R4
1455  006146  013705  001274                  MOV     SREG5,R5        ;RESTORE R5
1456  006152  000002                          RTI                     ;LEAVE
```

```
1457
1458                            ;       ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
1459                            ;       ;-----------------------------------------------------
1460                            ;       ;
1461  006154  104401  001313    .CONVR: TYPE    $CRLF
1462  006160  010046            .CNVRT: MOV     R0,-(SP)
1463  006162  010146                    MOV     R1,-(SP)
1464  006164  010346                    MOV     R3,-(SP)
1465  006166  010446                    MOV     R4,-(SP)
1466  006170  010546                    MOV     R5,-(SP)
1467  006172  017601  000012            MOV     @12(SP),R1
1468  006176  062766  000002  000012    ADD     #2,12(SP)
1469  006204  012137  006376            MOV     (R1)+,WRDCNT
1470  006210  112137  006400      1$:   MOVB    (R1)+,CHRCNT
1471  006214  112137  006401            MOVB    (R1)+,SPACNT
1472  006220  013137  006402            MOV     @(R1)+,BINWRD
1473  006224  122737  000003  006400    CMPB    #3,CHRCNT
1474  006232  001003                    BNE     2$
1475  006234  042737  177400  006402    BIC     #177400,BINWRD
1476  006242  013704  006402      2$:   MOV     BINWRD,R4
1477  006246  113705  006400            MOVB    CHRCNT,R5
1478  006252  012700  011112            MOV     #TEMP,R0
1479  006256  010403            3$:     MOV     R4,R3
1480  006260  042703  177770            BIC     #177770,R3
1481  006264  062703  000060            ADD     #060,R3
1482  006270  110320                    MOVB    R3,(R0)+
1483  006272  000241                    CLC
1484  006274  006004                    ROR     R4
1485  006276  000241                    CLC
1486  006300  006004                    ROR     R4
1487  006302  000241                    CLC
1488  006304  006004                    ROR     R4
1489  006306  005305                    DEC     R5
1490  006310  001362                    BNE     3$
1491  006312  012703  011154            MOV     #WDATA,R3
1492  006316  114023            4$:     MOVB    -(R0),(R3)+
1493  006320  105337  006400            DECB    CHRCNT
1494  006324  001374                    BNE     4$
1495  006326  105737  006401            TSTB    SPACNT
1496  006332  001405                    BEQ     6$
1497  006334  112723  000040      5$:   MOVB    #40,(R3)+
1498  006340  105337  006401            DECB    SPACNT
1499  006344  001373                    BNE     5$
1500  006346  105013            6$:     CLRB    (R3)
1501  006350  104401  011154            TYPE    ,WDATA
1502  006354  005337  006376            DEC     WRDCNT
1503  006360  001313                    BNE     1$
1504  006362  012605                    MOV     (SP)+,R5
1505  006364  012604                    MOV     (SP)+,R4
1506  006366  012603                    MOV     (SP)+,R3
1507  006370  012601                    MOV     (SP)+,R1
1508  006372  012600                    MOV     (SP)+,R0
1509  006374  000002                    RTI
1510  006376  000000  WRDCNT: 0
1511  006400  000000  CHRCNT: 0
1512  006401  SPACNT=CHRCNT+1
```

```
                        BINWRD: 0


                                ;TRAP DISPATCH SERVICE
                                ;ARGUMENT OF TRAP IS EXTRACTED
                                ;AND USED AS OFFSET TO OBTAIN POINTER
                                ;TO SELECTED SUBROUTINE

                        .SBTTL  TRAP DECODER

                        ;;*****************************************************************
                        ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                        ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                        ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                        ;*GO TO THAT ROUTINE.

006404 010046           STRAP:  MOV     R0,-(SP)        ;;SAVE R0
006406 016600 000002            MOV     2(SP),R0        ;;GET TRAP ADDRESS
006412 005740                   TST     -(R0)           ;;BACKUP BY 2
006414 111000                   MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
006416 006300                   ASL     R0              ;;POSITION FOR INDEXING
006420 016000 006440            MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
006424 000200                   RTS     R0              ;;GO TO ROUTINE


                        ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

006426 011646           STRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
006430 016666 000004 000002     MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
006436 000002                   RTI                     ;;RESTORE THE PSW


                        .SBTTL  TRAP TABLE

                        ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                        ;*BY THE "TRAP" INSTRUCTION.

                        ;       ROUTINE
                        ;       -------

006440 006426           $TRPAD: .WORD   STRAP2
006442 004404                   $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE


006444 005134                   $RDCHR  ;;CALL=RDCHR    TRAP+2(104402)  TTY TYPEIN CHARACTER ROUTINE
006446 005254                   $RDLIN  ;;CALL=RDLIN    TRAP+3(104403)  TTY TYPEIN STRING ROUTINE
006450 005554                   $RDOCT  ;;CALL=RDOCT    TRAP+4(104404)  READ AN OCTAL NUMBER FROM TTY
006452 004360                   .SCOP1  ;;CALL=SCOP1    TRAP+5(104405)  CALL TO LOOP ON CURRENT DATA HANDLER
006454 006042                   .SAVD5  ;;CALL=SAVD5    TRAP+6(104406)  CALL TO REGISTER SAVE ROUTINE
006456 006122                   .RESD5  ;;CALL=RESD5    TRAP+7(104407)  CALL TO REGISTER RESTORE ROUTINE
006460 007634                   .RSTCLR ;;CALL=RSTCLR   TRAP+10(104410) CALL TO ISSUE A MASTER CLEAR
006462 007304                   .DELAY  ;;CALL=DELAY    TRAP+11(104411) CALL TO DELAY
006464 007352                   .ROMCLK ;;CALL=ROMCLK   TRAP+12(104412) CALL TO CLOCK ROM ONCE
006466 007420                   .DATACLK        ;;CALL=DATACLK  TRAP+13(104413) CALL TO CLOCK DATA
006470 007464                   .TIMER  ;;CALL=TIMER    TRAP+14(104414) CALL TO DELAY A CLOCK TICK
006472 005714                   $INPUT  ;;CALL=INPUT    TRAP+15(104415) CALL TO OCTAL # INPUT ROUTINE
006474 006154                   .CONVRT ;;CALL=CONVRT   TRAP+16(104416) CALL TO     .....
006476 006160                   .CNVRT  ;;CALL=CNVRT    TRAP+17(104417) CALL TO     .......
```

```
1569  006500  006050                        .ADVANCE        ;;CALL=ADVANCE  TRAP+20(104420) CALL TO ADVANCE TO NEXT TEST
1570                                  ;
1571                                  ;-------------------------------------------------------------------
1572                                  ;:*****************************************************************
1573                                          ;ERROR HANDLER
1574                                          ;-------------
1575
1576  006502  004737  011216          SERROR: JSR     PC,CKSWR        ;CHECK FOR SOFT SWR
1577  006506  032777  010000  172524          BIT     #SW12,@SWR      ;BELL ON ERROR?
1578  006514  001406                          BEQ     XBX             ;BR IF NO BELL
1579  006516  105777  172526                  TSTB    @#TPS           ;TTY READY.
1580  006522  100003                          BPL     XBX             ;DON'T WAIT IF TTY NOT READY.
1581  006524  112777  000207  172520          MOVB    #207,@#TPB      ;PUSH A BELL AT THE TTY.
1582  006532  032777  020000  172500  XBX:    BIT     #SW13,@SWR      ;DELETE ERROR PRINT OUT?
1583  006540  001107                          BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
1584  006542  021637  001216                  CMP     (SP),SERRPC     ;WAS THIS ERROR FOUND LAST TIME?
1585  006546  001404                          BEQ     1$              ;BR IF YES
1586  006550  011637  001216                  MOV     (SP),SERRPC     ;RECORD BEING HERE
1587  006554  105037  001203                  CLRB    SERFLG          ;PREPARE HEADER
1588  006560  104406                  1$:     SAVDS                   ;SAVE ALL PROC REGISTERS
1589  006562  011605                          MOV     (SP),R5         ;GET THE PC OF ERROR
1590  006564  162705  000002                  SUB     #2,R5           ;GET ADDRESS OF TRAP CALL
1591  006570  011504                          MOV     (R5),R4         ;GET ERROR INSTRUCTION
1592  006572  110437  001214                  MOVB    R4,$ITEMB       ; COPY ERROR # FOR APT HANDLING
1593  006576  006304                          ASL     R4              ;MULT BY TWO
1594  006600  061504                          ADD     (R5),R4         ;DOUBLE IT
1595  006602  006304                          ASL     R4              ;MULT AGAIN
1596  006604  042704  177001                  BIC     #177001,R4      ;CLEAR JUNK
1597  006610  062704  001512                  ADD     #SERRTB,R4      ;GET POINTER
1598  006614  012437  006730                  MOV     (R4)+,ERRMSG    ;GET ERROR MESSAGE
1599  006620  012437  006742                  MOV     (R4)+,DATAHD    ;GET DATA HEADRER
1600  006624  011437  006754                  MOV     (R4),DATABP     ;GET DATA TABLE
1601  006630  105737  001203                  TSTB    SERFLG          ;TYPE HEADREER
1602  006634  001403                          BEQ     TYPMSG          ;BR IF YES
1603  006636  005737  006754                  TST     DATABP          ;DOES DATA TABLE EXIST?
1604  006642  001040                          BNE     TYPDAT          ;BR IF YES.
1605  006644  104401  001313          TYPMSG: TYPE    ,SCRLF
1606  006650  104401  001313                  TYPE    ,SCRLF
1607  006654  005737  001444                  TST     LOCK
1608  006660  001402                          BEQ     1$
1609  006662  104401  007767                  TYPE    ,MASTEK
1610  006666  104401  007755          1$:     TYPE    ,MTSTN
1611  006672  104417  007072                  CNVRT   ,XTSTN          ;SHOW IT
1612  006676  104401  010044                  TYPE    ,MERRPC         ;TYPE PC.
1613  006702  104417  007064                  CNVRT   ,ERTABD         ;SHOW IT
1614  006706  104401  001313                  TYPE    ,SCRLF          ;GIVE A CR/LF
1615  006712  112737  177777  001203          MOVB    #-1,SERFLG      ;NO MORE HEADER UNLESS NO DATA TABLE.
1616  006720  005737  006730                  TST     ERRMSG          ;IS THERE AN ERROR MESSAGE?
1617  006724  001402                          BEQ     WRKO.FM         ;BR IF NO.
1618  006726  104401                          TYPE                    ;TYPE
1619  006730  000000          ERRMSG: 0               ;    ERROR MESSAGE
1620  006732                  WRKO.FM:
1621  006732  005737  006742                  TST     DATAHD          ;DATA HEADER?
1622  006736  001402                          BEQ     TYPDAT          ;BR IF NO
1623  006740  104401                          TYPE                    ;TYPE
1624  006742  000000          DATAHD: 0               ;    DATA HEADER
```

```
1625  006744  005737  006754          TYPDAT: TST     DATABP          ;DATA TABLE?
1626  006750  001402                          BEQ     RESREG          ;BR IF NO.
1627  006752  104416                          CONVRT                  ;SHOW
1628  006754  000000                  DATABP: 0                       ;    DATA TABLE
1629  006756  104407                  RESREG: RESDS                   ;RESTORE PROC REGISTERS
1630  006760  122737  000001  001336  HALTS:  CMPB    @APTENV,$ENV    ; IS APT RUNNING ?
1631  006766  001007                          BNE     3$              ; SKIP APT CALL IF NOT.
1632  006770  113737  001214  007002          MOVB    $ITEMB,6$       ; COPY ERROR #.
1633  006776  004737  004704                  JSR     PC,$ATY4        ; CALL APT SERVICES.
1634  007002  000000                  6$:     .WORD   0               ; ERROR # GOES HERE.
1635  007004  000777                  9$:     BR      9$              ; LOCK HERE.
1636  007006  022737  004070  000042  3$:     CMP     @$ENDAD,#42     ;IF ACT-11 AUTOMATIC MODE, HALT!!
1637  007014  001403                          BEQ     1$
1638  007016  005777  172216                  TST     @SWR            ;HALT ON ERROR?
1639  007022  100005                          BPL     EXITER          ;BR IF NO HALT ON ERROR
1640  007024  010046                  1$:     PUSHRO                  ;SAVE R0
1641  007026  016600  000002                  MOV     2(SP),R0        ;SHOW ERROR PC IN DATA LIGHTS
1642  007032  000000                          HALT                    ;HALT
1643  007034  012600                          POPRO                   ;GET R0
1644  007036  005237  001212          EXITER: INC     SERTTL          ;UPDATE ERROR COUNT
1645  007042  032777  000400  172170          BIT     #SW08,@SWR      ;GOTO TOP OF TEST?
1646  007050  001004                          BNE     1$              ;BR IF YES
1647  007052  032777  002000  172160          BIT     #SW10,@SWR      ;GOTO NEXT TEST?
1648  007060  001400                          BEQ     2$              ;BR IF NO
1649                                          MOV     NEXT,$LPADR     ;SET FOR NEXT TEST
1650  007062                          1$:     ;MOV    #STACK,SP       ;RESET SP
1651                                  .       JMP     @$LPADR         ;GOTO SPECIFIED TEST
1652  007062  000002                  2$:     RTI                     ;$LPADR
1653  007064  000001                  ERTABO: 1
1654  007066    006   002                     .BYTE   6,2
1655  007070  001460                  ERTAB1: SAVPC
1656  007072  000001                  XTSTN:  1
1657  007074    003   002                     .BYTE   3,2
1658  007076  001202                          $TSTNM
1659                                          ;ENTER HERE ON POWER FAILURE
1660                                          ;--------------------------
1661
1662                                  .SBTTL  POWER DOWN AND UP ROUTINES
1663
1664                                  ;;**********************************************************************
1665                                  ;POWER DOWN ROUTINE
1666  007100  012737  007270  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
1667  007106  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
1668  007114  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
1669  007116  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
1670  007120  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
1671  007122  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
1672  007124  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
1673  007126  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
1674  007130  017746  172104                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
1675  007134  010637  007274                  MOV     SP,$SAVR6       ;;SAVE SP
1676  007140  012737  007152  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
1677  007146  000000                          HALT
1678  007150  000776                          BR      .-2             ;;HANG UP
1679
1680                                  ;;**********************************************************************
```

```
 1681                                   ;POWER UP ROUTINE
 1682  007152  012737  007270  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
 1683  007160  013706  007274                  MOV     $SAVR6,SP        ;;GET SP
 1684  007164  005037  007274                  CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
 1685  007170  005237  007274           1$:    INC     $SAVR6           ;;WAIT FOR THE INC
 1686  007174  001375                        BNE     1$               ;;OF  WORD
 1687  007176  104401  007534                  TYPE    ,MPFAIL          ;
 1688  007202  104417  007276                  CNVRT   ,PFTAB           ;
 1689  007206  105037  001203                  CLRB    $ERFLG           ;CLEAR ERROR FLAG.
 1690  007212  005037  001216                  CLR     $ERRPC           ; CLEAR LAST ERROR PC
 1691  007216  013701  002066                  MOV     KMCSR,R1         ; RESTORE DEVICE ADDRESS.
 1692  007222  005011                        CLR     (R1)             ; CLEAR THE CSR.
 1693  007224  104410                        MSTCLR                   ;
 1694  007226  012677  172006                  MOV     (SP)+,@SWR       ;;POP STACK INTO @SWR
 1695  007232  012605                        MOV     (SP)+,R5         ;;POP STACK INTO R5
 1696  007234  012604                        MOV     (SP)+,R4         ;;POP STACK INTO R4
 1697  007236  012603                        MOV     (SP)+,R3         ;;POP STACK INTO R3
 1698  007240  012602                        MOV     (SP)+,R2         ;;POP STACK INTO R2
 1699  007242  012601                        MOV     (SP)+,R1         ;;POP STACK INTO R1
 1700  007244  012600                        MOV     (SP)+,R0         ;;POP STACK INTO R0
 1701  007246  012737  007100  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
 1702  007254  012737  000340  000026        MOV     #340,@#PWRVEC+2  ;;PRIO:7
 1703  007262  104401                        TYPE                     ;;REPORT THE POWER FAILURE
 1704  007264  007534           $PWRMG: .WORD   MPFAIL                ;;POWER FAIL MESSAGE POINTER
 1705  007266  000002                        RTI
 1706  007270  000000           $ILLUP: HALT                          ;;THE POWER UP SEQUENCE WAS STARTED
 1707  007272  000776                        BR      .-2              ;;  BEFORE THE POWER DOWN WAS COMPLETE
 1708  007274  000000           $SAVR6: 0                             ;;PUT THE SP HERE
 1709
 1710  007276  000001           PFTAB:  1
 1711  007300     003     002           .BYTE   3,2
 1712  007302  001202                        $TSTNM
 1713
 1714  007304                   .DELAY:
 1715  007304  012777  000020  172562          MOV     #20,@KMP04
 1716  007312  104412                        ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 1717  007314  121111                        121111                   ;POKE CLOCK DELAY BIT
 1718  007316                           1$:
 1719  007316  104412                        ROMCLK                   ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 1720  007320  121224                        121224                   ;PORT4+IBUS*11
 1721  007322  032777  000020  172544          BIT     #BIT4,@KMP04     ;IS CLOCK BIT SET?
 1722  007330  001772                        BEQ     1$               ;BR IF NO
 1723  007332  000002                        RTI
 1724
 1725  007334                   .MSTCLR:
 1726  007334  152777  000100  172526          BISB    #BIT6,@KMCSRH    ;SET MASTER CLEAR
 1727  007342  142777  000300  172520          BICB    #BIT6!BIT7,@KMCSRH  ;CLEAR MASTER CLEAR AND RUN
 1728  007350  000002                        RTI                      ;RETURN
 1729
 1730  007352                   .ROMCLK:
 1731  007352  152777  000002  172510          BISB    #BIT1,@KMCSRH    ;SET ROMI
 1732  007360  013677  172512                  MOV     @(SP)+,@KMP06    ;LOAD INSTRUCTION IN SEL6
 1733  007364  062746  000002                  ADD     #2,-(SP)         ;ADJUST STACK
 1734  007370  032777  000100  171642          BIT     #SW06,@SWR       ;HALT IF SW06 =1
 1735  007376  001401                        BEQ     1$               ;BR IF SW06 =0
 1736  007400  000000                        HALT                     ;HALT BEFORE CLOCKING INSTRUCTION
```

```
 1737  007402  152777  000003  172460  1$:     BISB    #BIT1!BIT0,@KMCSRH ;CLOCK INSTRUCTION
 1738  007410  142777  000007  172452          BICB    #BIT2!BIT1!BIT0,@KMCSRH  ;CLEAR ROMO, ROMI, STEP
 1739  007416  000002                          RTI
 1740
 1741  007420                          .DATACLK:
 1742  007420  013637  011112                  MOV     @(SP)+,TEMP     ;PUT TICK COUNT IN TEMP
 1743  007424  062746  000002                  ADD     #2,-(SP)        ;ADJUST STACK
 1744  007430  152777  000020  172432  1$:     BISB    #BIT4,@KMCSRH   ;SET STEP LU
 1745  007436  027777  172424  172422          CMP     @KMCSR,@KMCSR   ;WASTE TIME
 1746  007444  142777  000020  172416          BICB    #BIT4,@KMCSRH   ;CLEAR STEP LU
 1747  007452  005337  011112                  DEC     TEMP            ;DEC TICK COUNT
 1748  007456  001364                          BNE     1$              ;BR IF NOT DONE
 1749  007460  000002                          RTI                     ;RETURN
 1750  007462  000001                  3$:     .BLKW 1
 1751
 1752  007464                          .TIMER:
 1753  007464  013637  011112                  MOV     @(SP)+,TEMP     ;MOVE COUNT TO TEMP
 1754  007470  062746  000002                  ADD     #2,-(SP)        ;ADJUST STACK
 1755  007474                          1$:
 1756  007474  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 1757  007476  021364                          021364                  ;PORT4+IBUS# REG11
 1758  007500  032777  000002  172366          BIT     #2,@KMP04       ;IS PGM CLOCK BIT CLEAR?
 1759  007506  001772                          BEQ     1$              ;BR IF YES
 1760  007510                          2$:
 1761  007510  104412                          ROMCLK                  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
 1762  007512  021364                          021364                  ;PORT4+IBUS# REG11
 1763  007514  032777  000002  172352          BIT     #2,@KMP04       ;IS PGM CLOCK BIT SET?
 1764  007522  001372                          BNE     2$              ;BR IF YES
 1765  007524  005337  011112                  DEC     TEMP            ;DEC COUNT
 1766  007530  001361                          BNE     1$              ;BR IF NOT DONE
 1767  007532  000002                          RTI                     ;RETURN
 1768
 1769  007534  050200  051127  043040  MPFAIL: .ASCIZ  <200>/PWR FAILED. RESTART AT TEST /
   (2) 007572  042600  042116  050040  MEPASS: .ASCIZ  <200>/END PASS DZKCA  /
   (2) 007614  051200     000  ...     MR:     .ASCIZ  <200>/R/
   (2) 007617     200  047516  042040  MEAR2:  .ASCIZ  <200>/NO DEVICES PRESENT./
   (2) 007644  044600  051516  043125  MEAR3:  .ASCIZ  <200>/INSUFFICIENT DATA!/
   (2) 007670  046200  041517  020113  MLOCK:  .ASCIZ  <200>/LOCK ON SELECTED TEST/
   (2) 007717     103  051123  020072  MCSRX:  .ASCIZ  /CSR: /
   (2) 007725     124  041505  020072  MVECX:  .ASCIZ  /VEC: /
   (2) 007733     120  051501  042523  MPASSX: .ASCIZ  /PASSES: /
   (2) 007744  051105  047522  051522  MERRX:  .ASCIZ  /ERRORS: /
   (2) 007755     124  051505  020124  MTSTN:  .ASCIZ  /TEST NO: /
   (2) 007767      52     000  ...     MASTEK: .ASCIZ  /*/
   (2) 007771  ...     042523  020124  MMEM:   .ASCIZ  <200>/SET SWITCH REG TO KMC11'S DESIRED ACTIVE./
   (2) 010044  041520  020072     000  MERRPC: .ASCIZ  /PC: /
   (2) 010051     200  020040  020040  XHEAD:  .ASCII  <200>/          MAP OF KMC11 STATUS/
   (2) 010110  020200  020040  020040          .ASCII  <200>/                  ----------/
   (2) 010147     200  020040  041520          .ASCII  <200>/  PC      CSR     STAT1    STAT2    STAT3/
   (2) 010221     200  042455  042455          .ASCIZ  <200>/------  ------  ------   ------   ------/
   (2) 010275     200  047510  020127  NUM:    .ASCIZ  <200>/HOW MANY KMC11'S TO BE TESTED?/
   (2) 010336     200  051503  020122  CSR:    .ASCIZ  <200>/CSR ADDRESS?/
   (2) 010353     200  042526  052103  VEC:    .ASCIZ  <200>/VECTOR ADDRESS?/
   (2) 010374  041200  020122  051120  PRIO:   .ASCIZ  <200>/BR PRIORITY LEVEL? (4,5,6,7)?/
   (2) 010433     200  044127  041511  MODU:   .ASCIZ  <200>/WHICH LINE UNIT? IF NONE TYPE "N", IF MB201 TYPE "1", IF MB202 TYP
   (2) 010545     200  053523  052111  LINE:   .ASCIZ  <200>/SWITCH PAC#1 (DDCMP LINE #)?/
```

```
(2) 010603      200  053523  052111  BM:     .ASCIZ  <200>/SWITCH PAC#2 (BM873 BOOT ADD)?/
(2) 010643      200  051511  052040  CONN:   .ASCIZ  <200>/IS THE LOOP BACK CONNECTOR ON?/
(2) 010703      200  047516  042040  NOACT:  .ASCIZ  <200>/NO DEVICES ARE SELECTED/
(2) 010734   100200  046513  030503  CONERR: .ASCIZ  <200><200>/KMC11 AT NONSTANDARD ADDRESS  PC: /
(2) 011001      200  054105  042520  CNERR:  .ASCIZ  <200>/EXPECTED  FOUND/
(2) 011022   024040  046513  024503  KMCM:   .ASCIZ  / (KMC) /
(2) 011032   044600  046114  043505  MILINT: .ASCIZ  <200>/ILLEGAL INTERRUPT ERROR/
(2)          011064                       .EVEN
(2) 011064   000005                  XSTATQ: 5
1770 011066      006      003                .BYTE   6,3
1771 011070   001276                        STMP0
1772 011072      006      003                .BYTE   6,3
1773 011074   001300                        STMP1
1774 011076      006      003                .BYTE   6,3
1775 011100   001302                        STMP2
1776 011102      006      003                .BYTE   6,3
1777 011104   001304                        STMP3
1778 011106      006      002                .BYTE   6,2
1779 011110   001306                        STMP4
1780                                  .EVEN
1781
1782                                  ;BUFFERS FOR INPUT-OUTPUT
1783
1784 011112   000000                  TEMP:   0
1785          011154                  .=.+40
1786 011154   000000                  MDATA:  0
1787          011216                  .=.+40
1788
1789
1790                                  ;ROUTINE USED TO CHANGE SOFTWARE SWITCH
1791                                  ;REGISTER USING THE CONSOLE TERMINAL
1792                                  ;-----------------------------------
1793
1794 011216   022737   000176  001240  CKSWR:  CMP     #SWREG,SWR      ;IS THE SOFT SWR BEING USED?
1795 011224   001075                          BNE     CKSWR5          ;BR IF NO
1796 011226   132737   000001  001336          BITB    #1,SENV         ; IS IT RUNNING UNDER APT?
1797 011234   001071                          BNE     CKSWR5          ; EXIT IF YES.
1798 011236   022777   000007  170002          CMP     #7,@STKB        ;WAS CTRL G TYPED? (7 BIT ASCII)
1799 011244   001404                          BEQ     1$              ;BR IF YES
1800 011246   022777   000207  167772          CMP     #207,@STKB      ;WAS CTRL G TYPED? (8 BIT ASCII)
1801 011254   001061                          BNE     CKSWR5          ;BR IF NO
1802 011256   010246                  1$:     MOV     R2,-(SP)        ;STORE R2
1803 011260   010346                          MOV     R3,-(SP)        ;STORE R3
1804 011262   010446                          MOV     R4,-(SP)        ;STORE R4
1805 011264   012737   177777  011422          MOV     #-1,SWFLG       ;SET SOFT TYPE OUT FLAG
1806 011272   005002                  CKSWR1: CLR     R2              ;CLEAR NEW SWR CONTENTS
1807 011274   012704   177777                  MOV     #-1,R4          ;SET FLAG TO ALL ONES
1808 011300   104401   005531                  TYPE    ,SMSWR          ;TYPE "SWR= "
1809 011304   104417                  CKSWR2: CNVRT                   ;TYPE OUT PRESENT CONTENTS
1810 011306   011456                          SOFTSW                  ;OF SOFT SWITCH REGISTER
1811 011310   104401   005542          CKSWR3: TYPE    ,SMNEW          ;TYPE "NEW? "
1812 011314   004737   011424          CKSWR4: JSR     PC,INCHAR       ;GET RESPONSE
1813 011320   022703   000015                  CMP     #15,R3          ;WAS IT A CR?
1814 011324   001424                          BEQ     5$              ;BR IF YES
1815 011326   022703   000012                  CMP     #12,R3          ;WAS IT A LF?
1816 011332   001416                          BEQ     4$              ;BR IF YES
```

```
1817  011334  022703  000025          CMP     #25,R3          ;WAS IT CTRL U?
1818  011340  001754                  BEQ     CKSWR1          ;BR IF YES(START OVER)
1819  011342  022703  000007          CMP     #7,R3           ;IF CNTL G GET NEXT CHAR
1820  011346  001762                  BEQ     CKSWR4
1821  011350  005004                  CLR     R4              ;IT MUST BE A DIGIT SO CLR FLAG
1822  011352  042703  177770          BIC     #177770,R3      ;ONLY 0-7 ARE LEGAL SO MASK OFF BITS
1823  011356  006302                  ASL     R2              ;SHIFT R2 3 TIMES
1824  011360  006302                  ASL     R2
1825  011362  006302                  ASL     R2
1826  011364  050302                  BIS     R3,R2           ;ADD LAST DIGIT
1827  011366  000752                  BR      CKSWR4          ;GET NEXT CHARACTER
1828  011370  012766  002402  000006  4$:     MOV     #.START,6(SP)   ;LF WAS TYPED SO GO TO START
1829  011376  005704          5$:     TST     R4              ;IS FLAG CLEAR?
1830  011400  001002                  BNE     6$              ;IF NOT DON'T CHANGE SOFT SWR
1831  011402  010277  167632          MOV     R2,@SWR         ;IF YES THEN WRITE NEW CONTENTS TO SOFT SWR
1832  011406  005037  011422  6$:     CLR     SWFLG           ;CLEAR TYPEOUT FLAG
1833  011412  012604                  MOV     (SP)+,R4        ;RESTORE R4
1834  011414  012603                  MOV     (SP)+,R3        ;RESTORE R3
1835  011416  012602                  MOV     (SP)+,R2        ;RESTORE R2
1836  011420  000207          CKSWR5: RTS     PC              ;RETURN
1837
1838  011422  000000          SWFLG:  0
1839
1840  011424  105777  167614  INCHAR: TSTB    @STKS
1841  011430  100375                  BPL     .-4
1842  011432  017703  167610          MOV     @STKB,R3
1843  011436  105777  167606          TSTB    @STPS
1844  011442  100375                  BPL     .-4
1845  011444  010377  167602          MOV     R3,@STPB
1846  011450  042703  000200          BIC     #BIT7,R3
1847  011454  000207                  RTS     PC
1848
1849  011456  000001          SOFTSW: 1
1850  011460  006       002           .BYTE   6,2
1851  011462  000176                  SWREG
```

```
1852
1853
1854                                    ;
1855                                    ;ROUTINE USED TO "CYCLE" THROUGH UP TO 16 KMC11'S
1856                                    ;THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC
1857                                    ;AND RUNS THE SPECIFIED KMC11'S.  THIS ROUTINE *MUST*
1858                                    ;BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE
1859                                    ;SETUP NECESSARY.
1860                                    ;
1861 011464 005737 001470      CYCLE:  TST     KMACTV          ;ARE ANY KMC11'S TO BE TESTED?
1862 011470 001004                     BNE     1$              ;BR IF OK.
1863 011472 104401 010703              TYPE    ,NOACT          ;NO KMC11'S SELECTED!!
1864 011476 000000                     HALT                    ;STOP THE SHOW.
1865 011500 000776                     BR      .-2             ;DISQUALIFY CONT. SW.
1866 011502 000241             1$:     CLC                     ;CLEAR PROC. CARRY BIT.
1867 011504 006137 001500              ROL     RUN             ;UPDATE POINTER
1868 011510 006537 001500              ADC     RUN             ;CATCH CARRY FROM RUN
1869 011514 062737 000004 001504       ADD     #4,MILK         ;UPDATE POINTER
1870 011522 062737 000010 001502       ADD     #10,CREAM       ;UPDATE ADDRESS POINTER.
1871 011530 022737 002300 001502       CMP     #KM.MAP+200,CREAM
1872 011536 001006                     BNE     2$              ;KEEP GOING; NOT ALL TESTED FOR.
1873 011540 012737 002100 001502       MOV     #KM.MAP,CREAM   ;RESET ADDRESS POINTER.
1874 011546 012737 002302 001504       MOV     #CNT.MAP,MILK   ;RESET PASS COUNT POINTER
1875 011554 033737 001500 001470 2$:   BIT     RUN,KMACTV      ;IS THIS ONE ACTIVE?
1876 011562 001747                     BEQ     1$              ;BR IF NO
1877 011564 013700 001502              MOV     CREAM,R0        ;GET ADDRESS POINTER
1878 011570 013702 001504              MOV     MILK,R2         ;GET PASS COUNT POINTER
1879 011574 012037 002066              MOV     (R0)+,KMCSR     ;LOAD SYSTEM CTRL. REG
1880 011600 011037 002056              MOV     (R0),KMRVEC     ;LOAD VECTOR
1881 011604 042737 177000 002056       BIC     #177000,KMRVEC  ;CLEAR UNWANTED BITS
1882 011612 012037 002050              MOV     (R0)+,STAT1     ;LOAD STAT1
1883 011616 012037 002052              MOV     (R0)+,STAT2     ;LOAD STAT2
1884 011622 012037 002054              MOV     (R0)+,STAT3     ;LOAD STAT3
1885 011626 012237 001364              MOV     (R2)+,SPASS     ;LOAD PASS COUNT
1886 011632 012237 001212              MOV     (R2)+,SERTTL    ;LOAD ERROR COUNT
1887 011636 012700 000002              MOV     #2,R0           ;SAVE CORE THIS WAY!
1888 011642 013737 002066 002070       MOV     KMCSR,KMCSRH
1889 011650 005237 002070              INC     KMCSRH
1890 011654 013737 002070 002072       MOV     KMCSRH,KMCTL
1891 011662 005237 002072              INC     KMCTL
1892 011666 013737 002072 002074       MOV     KMCTL,KMP04
1893 011674 060037 002074              ADD     R0,KMP04
1894 011700 013737 002074 002076       MOV     KMP04,KMP06
1895 011706 060037 002076              ADD     R0,KMP06
1896
1897 011712 013737 002056 002060       MOV     KMRVEC,KMRLVL   ;PTY LVL
1898 011720 060037 002060              ADD     R0,KMRLVL       ;
1899 011724 013737 002060 002062       MOV     KMRLVL,KMTVEC   ;TX VEC
1900 011732 060037 002062              ADD     R0,KMTVEC       ;
1901 011736 013737 002062 002064       MOV     KMTVEC,KMTLVL   ;TX LVL
1902 011744 060037 002064              ADD     R0,KMTLVL
1903 011750 012777 034664 170100       MOV     #INISR,@KMRVEC  ;SET UP INTERRUPT VECTORS.
1904 011756 012777 000240 170074       MOV     #PRS,@KMRLVL    ;SET PRIORITY 5.
1905 011764 012777 035360 170070       MOV     #QUISR,@KMTVEC  ;SET UP INTERRUPT VECTORS.
1906 011772 012777 000240 170064       MOV     #PRS,@KMTLVL    ;SET PRIORITY 5.
1907 012000 012737 000200 177776       MOV     #PR4,PS         ;SET MAIN PROGRAM PRIORITY AS 4.
```

```
                                                                ;SOTHAT KMC CAN INTERRUPT...

012006  032737  000002  001446        BIT     #SW01,STRTSW    ;IS TEST NO. SELECTED
012014  001447                        BEQ     7$              ;BR IF NO
012016                          4$:
012016  005737  000042                TST     @#42            ;RUNNING IN AUTO MODE?
012022  001044                        BNE     7$              ;BR IF YES
012024  104401  001313                TYPE    ,SCRLF
012030  104415                        INPUT
012032  007755                        MTSTN
012034  000001                        1
012036  001000                        1000
012040  001202                        STSTNM
012042      000                 .BYTE   0
012043      001                 .BYTE   1
012044  012700  013774                MOV     #TST1,R0
012050  022710                  5$:   CMP     (PC)+,(R0)      ;CMP FIRST WORD TO 12737
012052  012737                        MOV     (PC)+,@(PC)+
012054  001020                        BNE     6$              ;BR IF NOT SAME
012056  023760  001202  000002        CMP     STSTNM,2(R0)    ;DOES STSTNM MATCH?
012064  001014                        BNE     6$              ;BR IF NO
012066  022760  001202  000004        CMP     #STSTNM,4(R0)   ;IS LAST WORD OK?
012074  001010                        BNE     6$              ;BR IF NO
012076  010037  001206                MOV     R0,SLPADR       ;IT IS A LEGAL TEST SO DO IT
012102  104401  007614                TYPE    ,MR
012106  042737  000002  001446        BIC     #SW01,STRTSW
012114  000412                        BR      8$
012116  005720                  6$:   TST     (R0)+           ;POP R0
012120  020027  034324                CMP     R0,#TLAST+10    ;AT END YET?
012124  001361                        BNE     5$              ;BR IF NO
012126  104401  001312                TYPE    ,$QUES          ;YES ILLEGAL TEST NO.
012132  000731                        BR      4$              ;TRY AGAIN

012134  012737  013774  001206  7$:   MOV     #TST1,SLPADR    ;PREPARE SLPADR ADDRESS
012142  013701  002066          8$:   MOV     KMCSR,R1        ;R1 = BASE KMC11 ADDRESS
012146  000177  167034                JMP     @SLPADR         ;GO START TESTING.



                                ;ROUTINE USED TO "AUTO SIZE" THE KMC11
                                ;CSR AND VECTOR.
                                ;NOTE:  THE CSR MAY BE ANY WHERE IN THE FLOATING
                                ;       ADDRESS RANGE (160000:164000)
                                ;       AND THE VECTOR MAY BE ANY WHERE IN THE
                                ;       FLOATING VECTOR RANGE (300:770)
                                ;

012152                  AUTO.SIZE:
012152  000005                  RESET                           ;INSURE A BUS INIT.
012154  012702  002100  CSRMAP: MOV     #KM.MAP,R2      ;LOAD MAP POINTER.
012160  005022          1$:     CLR     (R2)+           ;ZERO ENTIRE MAP
012162  022702  002300          CMP     #KM.END,R2      ;ALL DONE?
012166  001374                  BNE     1$              ;BR IF NO
012170  005037  001472          CLR     KMNUM           ;SET OCTAL NUMBER OF KMC11'S TO 0
012174  012702  002100          MOV     #KM.MAP,R2      ;R2 POINTS TO KMC MAP
012200  005037  001470          CLR     KMACTV          ;CLEAR ACTIVE
012204  032737  000001  001446  BIT     #SW00,STRTSW    ;QUESTIONS?
```

```
1964  012212  001002                  BNE     .+6             ;BR IF YES
1965  012214  000137  012574          JMP     @#S             ;IF NO SKIP QUESTIONS
1966  012220  012737  000001  001306  MOV     #1,STMP4        ;START WITH 1
1967  012226  104415                  INPUT
1968  012230  010275                  NUM
1969  012232  000001                  1
1970  012234  000020                  16.
1971  012236  001312                  STMP2
1972  012240    000                   .BYTE   0
1973  012241    001                   .BYTE   1
1974  012242  013737  001302  001472  MOV     STMP2,KMNUM     ;KMNUM = HOW MANY
1975  012250  104401  001313    12S:  TYPE    ,SCRLF
1976  012254  104416                  CONVRT                  ;TYPE WHICH KMC IS BEING DONE
1977  012256  013226                  WHICH                   ;STMP4 IS WHICH KMC
1978  012260  005237  001306          INC     STMP4
1979  012264  104415                  INPUT
1980  012266  010346                  CSR
1981  012270  160000                  160000
1982  012272  164000                  164000
1983  012274  001304                  STMP3
1984  012276    000                   .BYTE   0
1985  012277    001                   .BYTE   1
1986  012300  013722  001304          MOV     STMP3,(R2)+     ;STORE CSR IN MAP
1987  012304  104415                  INPUT
1988  012306  010353                  VEC
1989  012310  000000                  0
1990  012312  000776                  776
1991  012314  001304                  STMP3
1992  012316    000                   .BYTE   0
1993  012317    001                   .BYTE   1
1994  012320  013712  001304          MOV     STMP3,(R2)      ;STORE VECTOR IN MAP
1995  012324  104401    10S:  TYPE
1996  012326  010374                  PRIO                    ;ASK WHAT BR LEVEL
1997  012330  004737  013520          JSR     PC,INTTY        ;GET RESPONSE
1998  012334  022703  000024          CMP     #24,R3          ;
1999  012340  101014                  BHI     50S             ;BR IF LESS THAN 4
2000  012342  022703  000027          CMP     #27,R3          ;
2001  012346  103411                  BLO     50S             ;BR IF GREATER THAN 7
2002  012350  012704  000011          MOV     #11,R4          ;R4 = NUMBER OF SHIFTS
2003  012354  006303                  ASL     R3              ;SHIFT R3 LEFT
2004  012356  005304                  DEC     R4              ;DEC SHIFT COUNT
2005  012360  001375                  BNE     .-4             ;BR IF NOT DONE
2006  012362  042703  170777          BIC     #170777,R3      ;BIC UNWANTED BITS
2007  012366  050312                  BIS     R3,(R2)         ;PUT BR LEVEL IN STATUS MAP
2008  012370  000403                  BR      8S              ;CONTINUE
2009  012372  104401    50S:  TYPE
2010  012374  001312                  SQUES                   ;RESPONSE IS OUT OF LIMITS
2011  012376  000752                  BR      10S             ;TRY AGAIN
2012  012400            8S:
2013  012400            9S:
2014  012400  104401    16S:  TYPE
2015  012402  010433                  MODU                    ;ASK WHICH LINE UNIT
2016  012404  004737  013520          JSR     PC,INTTY        ;GET REPLY
2017  012410  022703  000021          CMP     #21,R3          ;"1"
2018  012414  001417                  BEQ     30S
2019  012416  022703  000022          CMP     #22,R3          ;"2"
```

```
                                        BEQ     31$
                000116                  CMP     #116,R3         ;"N"
                                        BEQ     32$
                                        TYPE
                                        SQUES                   ;IF NOT A 1,2 OR N TYPE "?"
                                        BR      16$             ;TRY AGIAN
                010000          32$:    BIS     #BIT12,(R2)+    ;SET BIT 12 IN STAT2 IF NO LU
                                        CMP     (R2)+,(R2)+     ;POP OVER STAT2 AND STAT3
                                        BR      33$
                020000          31$:    BIS     #BIT13,(R2)     ;SET BIT 13 IN STAT2 IF MB202
                                30$:    TYPE
                                        CONN                    ;ASK IF LOOP-BACK IS ON
                013520                  JSR     PC,INTTY        ;GET REPLY
                000131                  CMP     #131,R3         ;Y
                                        BEQ     17$
                000116                  CMP     #116,R3         ;N
                                        BEQ     18$
                                        TYPE
                                        SQUES                   ;IF NOT Y OR N TYPE "?"
                                        BR      30$             ;TRY AGAIN
                040000          17$:    BIS     #BIT14,(R2)+    ;TURNAROUND IS CONNECTED
                                        BR      19$
                040000          18$:    BIC     #BIT14,(R2)+    ;NO TURNAROUND
                                19$:
                                        INPUT
                                        LINE
                                        0
                                        377
                                        STMP3
                                        .BYTE   0
                                        .BYTE   1
                001304                  MOVB    STMP3,(R2)+     ;STORE SWITCH PAC IN MAP
                                        INPUT
                                        BM
                                        0
                                        377
                                        STMP3
                                        .BYTE   0
                                        .BYTE   1
                001304                  MOVB    STMP3,(R2)+     ;STORE SWITCH PAC IN MAP
                                        TST     (R2)+           ;POP OVER STAT3
                001302          33$:    DEC     STMP2           ;DEC KMC COUNT
                                        BNE     12$             ;BR IF MORE TO DO
                013126                  JMP     13$             ;CONTINUE
                160000          7$:     MOV     #160000,R1      ;SET FOR FIRST ADDRESS TO BE TESTED
                000004                  MOV     #6$,2#4         ;SET FOR NON-EXISTANT DEVICE TIME OUT
                                2$:     CLR     (R1)            ;CLEAR SEL0
                                        TST     (R1)            ;IF KMC11 KMCSR S/B 0
                                        BNE     3$              ;IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO KMC11
                000006                  CLR     6(R1)           ;CLEAR SEL6
                000006                  TST     6(R1)           ;IF KMC11 THEN KMRIC S/B =0!
                                        BNE     3$              ;BR IF NOT KMC11
                002000                  MOV     #BIT10,(R1)     ;SET ROMO
                000004                  CLR     4(R1)           ;CLEAR SEL4
                125252          000006  MOV     #125252,6(R1)   ;WRITE THIS TO SEL6
                020000                  BIS     #BIT13,(R1)     ;WRITE IT!
```

```
2076  012650  022761  125252  000004          CMP     #125252,4(R1)   ;WAS IT WRITTEN?
2077  012656  001113                          BNE     3$              ;IF NO IT IS NOT CRAM
2078                                  ;AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A KMC11 CSR ADDRESS.
2079  012660                          21$:
2080  012660  010122                  22$:    MOV     R1,(R2)+        ;STORE CSR IN CORE TABLE.
2081  012662  012711  001000          15$:    MOV     #BIT9,(R1)      ;CLEAR LINE UNIT LOOP
2082  012666  005061  000004                  CLR     4(R1)           ;CLEAR PORT4
2083  012672  012761  122113  000006          MOV     #122113,6(R1)   ;LOAD INSTRUCTION (CLR DTR)
2084  012700  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION
2085  012704  012761  021264  000006          MOV     #021264,6(R1)   ;LOAD INSTRUCTION
2086  012712  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION
2087  012716  122761  000377  000004          CMPB    #377,4(R1)      ;IS IT ALL ONES?
2088  012724  001003                          BNE     .+10            ;BR IF NO
2089  012726  052712  010000                  BIS     #BIT12,(R2)     ;IF YES, NO LINE UNIT, SET STATUS BIT
2090  012732  000436                          BR      20$
2091  012734  032761  000002  000004          BIT     #BIT1,4(R1)     ;IS SWITCH A ONE?
2092  012742  001403                          BEQ     .+10            ;BR IF M8201
2093  012744  052712  060000                  BIS     #BIT13!BIT14,(R2) ;M8202 ASSUME CONNECTOR
2094  012750  000427                          BR      20$             ;CONNECTOR ON)
2095  012752  032761  000010  000004          BIT     #BIT3,4(R1)     ;IS MRDY SET
2096  012760  001023                          BNE     20$             ;BR IF M8201 NO CONNECTOR (ON LINE)
2097  012762  012761  000100  000004          MOV     #BIT6,4(R1)     ;LOAD PORT4
2098  012770  012761  122113  000006          MOV     #122113,6(R1)   ;LOAD INSTRUCTION
2099  012776  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION(SET DTR)
2100  013002  012761  021264  000006          MOV     #021264,6(R1)   ;LOAD INSTRUCTION
2101  013010  052711  000400                  BIS     #BIT8,(R1)      ;CLOCK INSTRUCTION(READ MODEM REG)
2102  013014  032761  000010  000004          BIT     #BIT3,4(R1)     ;IS MRDY SET NOW?
2103  013022  001402                          BEQ     20$             ;BR IF NO CONNECTOR
2104  013024  052712  040000                  BIS     #BIT14,(R2)     ;SET STATUS BIT FOR CONNECTOR
2105  013030  005726                  20$:    TST     (R2)+           ;POP POINTER
2106  013032  012761  021324  000006          MOV     #021324,6(R1)   ;PUT INSTRUCTION IN PORT6
2107  013040  012711  001400                  MOV     #BIT9!BIT8,(R1) ;PORT4=LU 15
2108  013044  156122  000004                  BISB    4(R1),(R2)+     ;STORE DDCMP LINE # IN TABLE
2109  013050  012761  021344  000006          MOV     #021344,6(R1)   ;PORT6=INSTRUCTION
2110  013056  012711  001400                  MOV     #BIT8!BIT9,(R1) ;CLOCK INSTR.
2111  013062  156122  000004                  BISB    4(R1),(R2)+     ;STORE M8873 ADD IN TABLE
2112  013066  005726                          TST     (R2)+           ;POP OVER STAT3
2113  013070  005011                          CLR     (R1)            ;CLEAR MOM1
2114  013072  005237  001472                  INC     KMNUM           ;UPDATE DEVICE COUNTER
2115  013076  022737  000020  001472          CMP     #20,KMNUM       ;ARE MAX. NO. OF DEV FOUND?
2116  013104  001410                          BEQ     13$             ;YES DON'T LOOK FOR ANY MORE.
2117  013106  005011                  35:     CLR     (R1)            ;CLEAR BIT 10
2118  013110  005061  000006                  CLR     6(R1)           ;CLEAR SEL 6
2119  013114  062701  000010          14$:    ADD     #10,R1          ;UPDATE CSR POINTER ADDRESS
2120  013120  022701  164000                  CMP     #164000,R1
2121  013124  001230                          BNE     2$              ;BR IF MORE ADDRESS TO CHECK.
2122  013126  005037  001470          13$:    CLR     KMACTV
2123  013132  005737  001472                  TST     KMNUM           ;HERE ANY KMC11'S FOUND AT ALL?
2124  013136  001423                          BEQ     5$              ;ERROR AUTO SIZER FOUND NO KMC11'S IN THIS SYS.
2125  013140  013701  001472                  MOV     KMNUM,R1
2126  013144  010137  001476                  MOV     R1,SAVNUM       ;SAVE NUMBER OF DEVICES
2127  013150  000241                  4$:     CLC
2128  013152  006137  001470                  ROL     KMACTV          ;GENERATE ACTIVE REGISTER OF DEVICES.
2129  013156  005237  001470                  INC     KMACTV          ;SET THE BIT
2130  013162  005301                          DEC     R1
2131  013164  001371                          BNE     4$              ;BR IF MORE TO GENERATE
```

```
2132  013166  012737  000006  000004          MOV     #6,@#4          ;RESTORE TRAP VECTOR
2133  013174  013737  001470  001474          MOV     KMACTV,SAVACT   ;SAVE ACTIVE REGISTER
2134  013202  000137  013234                  JMP     VECMAP          ;GO FIND THE VECTOR NOW.
2135  013206  104401  007617          5$:     TYPE    ,MERR2          ;NOTIFY OPR THAT NO KMC11'S FOUND.
2136  013212  005000                          CLR     R0              ;MAKE DATA LIGHTS ZERO
2137  013214  000000                          HALT                    ;STOP THE SHOW
2138  013216  000776                          BR      .-2             ;DISABLE CONT. SW.
2139  013220  012716  013114          6$:     MOV     #14$,(SP)       ;ENTERED BY NON-EXISTANT TIME-OUT.
2140  013224  000002                          RTI                     ;RETURN TO MAINSTREAM
2141
2142  013226  000001          WHICH:  1
2143  013230      002     002                 .BYTE   2,2
2144  013232  001306                          STMP4
2145
2146  013234  032737  000001  001446  VECMAP: BIT     #SW00,STRTSW
2147  013242  001114                          BNE     5$
2148  013244  012737  000340  000022          MOV     #340,@#22       ;SET IOT TRAP PRIO TO 7
2149  013252  012737  013426  000020          MOV     #14$,@#20       ;SET IOT TRAP VECTOR
2150  013260  012702  002100                  MOV     #KM.MAP,R2      ;SET SOFTWARE POINTER
2151  013264  012700  000300                  MOV     #300,R0         ;FLOATING VECTORS START HERE.
2152  013270  012701  000302                  MOV     #302,R1         ;PC OF IOT INSTR.
2153  013274  010120          1$:     MOV     R1,(R0)+        ;START FILLING VECTOR AREA
2154  013276  012721  000004                  MOV     #4,(R1)+        ;WITH .+2; IOT
2155  013302  060021                          CMP     (R0)+,(R1)+     ;ADD 2 TO R0 +R1
2156  013304  020127  001000                  CMP     R1,#1000
2157  013310  101771                          BLOS    1$              ;BR IF MORE TO FILL
2158  013312  013737  001470  001276          MOV     KMACTV,STMP0    ;STORE TEMPORALLY
2159  013320  006037  001276          2$:     ROR     STMP0           ;BRING OUT A BIT
2160  013324  103003                          BCC     5$              ;BR IF ALL DONE
2161  013326  012704  000012                  MOV     #12,R4          ;R4 IS INDEX REGISTER
2162  013332  016437  013504  177776          MOV     BRLVL(R4),PS    ;SET PS TO 7
2163  013340  011201                          MOV     (R2),R1
2164  013342  012761  000200  000004          MOV     #200,4(R1)
2165  013350  012711  001000                  MOV     #BIT9,(R1)      ;SET ROMI
2166  013354  012761  121111  000006          MOV     #121111,6(R1)   ;PUT INSTRUCTION IN PORT6
2167  013362  012711  001400                  MOV     #BIT9!BIT8,(R1) ;FORCE AN INTERRUPT
2168  013366  105200          7$:     INCB    R0              ;STALL
2169  013370  001376                          BNE     .-2             ;FOR TIME TO INTERUPT
2170  013372  162704  000002                  SUB     #2,R4           ;GET NEXT LOWEST PS LEVEL
2171  013376  001404                          BEQ     6$              ;BR IF R4 = 0
2172  013400  016437  013504  177776          MOV     BRLVL(R4),PS    ;MOVE NEXT LOWER LEVEL IN PS
2173  013406  000767                          BR      7$              ;BR TO DELAY
2174  013410  052762  005300  000002  6$:     BIS     #5300,2(R2)     ;NO INTERUPT ASSUME 300 AT LEVEL 5 AND FIX KMC11 LATER
2175  013416  005011          3$:     CLR     (R1)            ;CLEAR ROMI
2176  013420  062702  000010                  ADD     #10,R2          ;POP SOFTWARE POINTER
2177  013424  000736                          BR      2$              ;KEEP GOING
2178  013426  051662  000002          4$:     BIS     (SP),2(R2)      ;GET VECTOR ADDRESS
2179  013432  047562  000007  000002          BIC     #7,2(R2)        ;CLEAR JUNK
2180  013440  016405  013506                  MOV     BRLVL+2(R4),R5  ;GET BR LEVEL OF KMC11
2181  013444  006305                          ASL     R5              ;SHIFT LEVEL 4 PLACES
2182  013446  006305                          ASL     R5              ;TO THE LEFT FOR THE
2183  013450  006305                          ASL     R5              ;STATUS TABLE
2184  013452  006305                          ASL     R5
2185  013454  042705  170777                  BIC     #170777,R5      ;CLEAR UNWANTED BITS
2186  013460  050562  000002                  BIS     R5,2(R2)        ;PUT BR LEVEL IN STATUS TABLE
2187  013464  022626                          CMP     (SP)+,(SP)+     ;POP IOT JUNK OFF STACK
```

```
                                        ;SET FOR RETURN
                                        ; RESTORE SCOPE VECTOR
                                        ;ALL DONE WITH "AUTO SIZING"

                                        ;LEVEL 0
                                        ;LEVEL 0
                                        ;LEVEL 4
                                        ;LEVEL 5
                                        ;LEVEL 6
                                        ;LEVEL 7

                                        ;WAIT FOR DONE
                                        ;PUT CHAR IN R3
                                        ;WAIT UNTIL PRINTER IS READY
                                        ;ECHO CHAR
                                        ;MASK OFF LOWER CASE
                                        ;RETURN

                                        ;;PUSH R0 ON STACK
                                        ;;PUSH R1 ON STACK
                                        ;;PUSH R2 ON STACK
                                        ;;PUSH R3 ON STACK
                                        ;;CLEAR THE LOCAL VARIABLE
                                        ;;CLEAN UP LOCAL VARIABLE
                                        ;;GET THE DEVICE COUNT
                                        ;;SAVE THE NO. OF DEVICES
                                        ;;GET EXTRA INFO. BITS POINTER
                                        ;;GET BASE CSR ADDRESS
                                        ;;GET THE VECTOR
                                        ;;GET THE PRIORITY
                                        ;;SAVE THE KMC'S SELECTED ACTIVE
                                        ;;SAVE THE ACTIVE REGISTER
                                        ;;GET ADDRESS OF FIRST DEVICE DESCRIPTOR WORD
                                        ;;GET POINTER TO DEVICE MAP
                                        ;;CLEAR DEVICE MAP
                                        ;;IS WHOLE DEV.MAP CLEARED?
                                        ;;NO, THEN GO ON.
                                        ;;RESTORE DEV.MAP POINTER.
                                        ;;LOAD CSR ADDRESS
                                        ;;GET EXTRA INFO. BITS
                                        ;;SET IT IN RIGHT POSITION.
                                        ;;SET IT IN RIGHT POSITION.
                                        ;;GET PRIORITY IN STAT1
                                        ;;SET THEM IN RIGHT POSITION
                                        ;;  "   "   "   "   "
                                        ;;  "   "   "   "   "
                                        ;;  "   "   "   "   "
                                        ;;GET THE VECTOR IN STAT1.
                                        ;;GET THE STAT2 FROM DDWXX
                                        ;;SKIP OVER STAT3
```

```
2244  013732  005300                        DEC     R0              ; COUNT BY 1
2245  013734  001407                        BEQ     2$              ; ALL DONE?
2246  013736  062737  000010  013770        ADD     #10,BASE        ; INCREMENT BASE CSR ADDRESS BY 10
2247  013744  062737  000010  013766        ADD     #10,VECTR       ; INCREMENT VECTOR ADDRESS BY 10
2248  013752  000747                        BR      1$              ; SET THE NEXT MAP ENTRY
2249  013754                        2$:
2250  013754  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
2251  013756  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
2252  013760  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
2253  013762  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
2254  013764  000207                        RTS     PC              ; RETURN
2255  013766  000000                VECTR:  .WORD   0
2256  013770  000000                BASE:   .WORD   0
2257  013772  000000                PRIRTY: .WORD   0
2258  013774                        $FLT1   OUT1,INP1,4,0
2259  013774                        $XZ
2260
2261
2262                                ;************************** TEST 1 **************************
2263                                ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
2264                                ;* FLOAT A 1 THROUGH REGISTER OUT1 (4)
2265                                ;* FLOAT A 0THROUGH REGISTER OUT1 (4)
2266  013774                        $XZ
2267                                ;;*********************************************************
2268
2269  013774                        $TSTN
2270                                ;  TEST 1
2271                                ;-------------
2272  013774  012737  000001  001202  TST1:  MOV     #1,$TSTNM               ; LOAD THE NO. OF THIS TEST
2273  014002  012737  014200  001442          MOV     #TST2,NEXT              ; POINT TO THE START OF NEXT TEST.
2274                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
2275  014010  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
2276  014014  014030                          MCT1
2277  014016  104022                          ERROR   22                      ;TIME OUT ERROR...
2278  014020  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
2279  014024  000177  165412                  JMP     @NEXT                   ;GO TO NEXT TEST...
2280  014030                        MCT1:
2281  014030                        21$:
2282  014030                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
              000000                            MICPC=MICPC+1
2284  014030  000400                          .WORD   .$.
2285  014032                                  MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
              000001                            MICPC=MICPC+1
2287  014032  063236                          .WORD   .$.
2288  014034                        $FLOT   OUT1,INP1,4,0,1,1$,2$,3$,4$
2289  014034                        1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
              000002                            MICPC=MICPC+1
2291  014034  000600                          .WORD   .$.
2292  014036                        2$:
2293  014036                                  MOVE    BREG,OUT1 (4)   ;SET THE BIT.
              000003                            MICPC=MICPC+1
2295  014036  061224                          .WORD   .$.
2296  014040                                  MOVE    INP1 (4),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
              000004                            MICPC=MICPC+1
2298  014040  123100                          .WORD   .$.
2299  014042                                  MOVE    BREG,SPAD (4)
```

```
                                MICPC=MICPC+1
                                .WORD   .$.
                                SIFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...
                                SUB2C   SPAD <0>,BREG,NOP
                                MICPC=MICPC+1
                                .WORD   .$.
                                BZ      3$
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    IMP1 <4>,OUT1 <CSR5>;BAD DATA...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 1,MEM                 ;TYPE OF ERROR...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    MEM,OUT1 <CSR3>         ;
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 4,MEM         ;
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
                                MICPC=MICPC+1
                                .WORD   .$.
                                CALL    ERROR                   ;REPORT DATA ERROR.
                                MOVE    # <MICPC+3>,BREG
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     ERROR
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     2$                      ;LOOP ON ERROR...
                                MICPC=MICPC+1
                                .WORD   .$.
                        3$:     CALL    SCP1                    ;IS LOOP DATA SET.
                                MOVE    # <MICPC+3>,BREG
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     SCP1
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    SPAD <4>,BREG
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     2$                      ;YES, DO IT.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    SPAD <4>,BREG
```

```
2356                000026                  MICPC=MICPC+1
2357  014104  060604                        .WORD   .$.
2358  014106                                SHFBRT                          ;NO, CONTINUE...
2359                000027                  MICPC=MICPC+1
2360  014106  061620                        .WORD   .SBR!.SELB!.DBRSH
2361  014110                                BB7     4$                      ;IS IT DONE?...
2362                000030                  MICPC=MICPC+1
2363  014110  103432                        .WORD   .$.
2364  014112                                SBR     2$                      ;NO, CONTINUE...
2365                000031                  MICPC=MICPC+1
2366  014112  100403                        .WORD   .$.
2367  014114                        4$:
2368  014114                        SFLOT   OUT1,INP1,4,0,0,11$,12$,13$,14$
2369  014114                        11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
2370                000032                  MICPC=MICPC+1
2371  014114  000577                        .WORD   .$.
2372  014116                        12$:
2373  014116                                MOVE    BREG,OUT1 <4>   ;SET THE BIT.
2374                000033                  MICPC=MICPC+1
2375  014116  061224                        .WORD   .$.
2376  014120                                MOVE    INP1 <4>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
2377                000034                  MICPC=MICPC+1
2378  014120  123100                        .WORD   .$.
2379  014122                                MOVE    BREG,SPAD <4>
2380                000035                  MICPC=MICPC+1
2381  014122  063224                        .WORD   .$.
2382  014124                                SIFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...
2383
2384
2385  014124                                SUB2C   SPAD <0>,BREG,NOP
2386                000036                  MICPC=MICPC+1
2387  014124  060360                        .WORD   .$.
2388  014126                                BZ      13$
2389                000037                  MICPC=MICPC+1
2390  014126  101452                        .WORD   .$.
2391  014130                                MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
2392                000040                  MICPC=MICPC+1
2393  014130  061224                        .WORD   .$.
2394  014132                                MOVE    INP1 <4>,OUT1 <CSR5>;BAD DATA...
2395                000041                  MICPC=MICPC+1
2396  014132  121105                        .WORD   .$.
2397  014134                                MOVE    # 1,MEM                 ;TYPE OF ERROR...
2398                000042                  MICPC=MICPC+1
2399  014134  002401                        .WORD   .$.
2400  014136                                MOVE    MEM,OUT1 <CSR3>         ;
2401                000043                  MICPC=MICPC+1
2402  014136  041223                        .WORD   .$.
2403  014140                                MOVE    # 4,MEM         ;
2404                000044                  MICPC=MICPC+1
2405  014140  002404                        .WORD   .$.
2406  014142                                MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
2407                000045                  MICPC=MICPC+1
2408  014142  041227                        .WORD   .$.
2409  014144                                CALL    ERROR                   ;REPORT DATA ERROR.
2410  014144                                MOVE    # <MICPC+3>,BREG
2411                000046                  MICPC=MICPC+1
```

```
2412  014144  000450          .WORD   .$.
2413  014146                  SBR     ERROR
2414          000047          MICPC=MICPC+1
2415  014146  104400          .WORD   .$.
2416  014150                  MOVE    SPAD (4),BREG           ;RESTORE BREG...
2417          000050          MICPC=MICPC+1
2418  014150  060604          .WORD   .$.
2419  014152                  SBR     12$                     ;LOOP ON ERROR...
2420          000051          MICPC=MICPC+1
2421  014152  100433          .WORD   .$.
2422  014154          13$:    CALL    SCP1                    ;IS LOOP DATA SET.
2423  014154                  MOVE    @ (MICPC+3),BREG
2424          000052          MICPC=MICPC+1
2425  014154  000454          .WORD   .$.
2426  014156                  SBR     SCP1
2427          000053          MICPC=MICPC+1
2428  014156  104427          .WORD   .$.
2429  014160                  MOVE    SPAD (4),BREG
2430          000054          MICPC=MICPC+1
2431  014160  060604          .WORD   .$.
2432  014162                  SBR     12$                     ;YES, DO IT.
2433          000055          MICPC=MICPC+1
2434  014162  100433          .WORD   .$.
2435  014164                  MOVE    SPAD (4),BREG
2436          000056          MICPC=MICPC+1
2437  014164  060604          .WORD   .$.
2438  014166                  SHFBRT                          ;NO, CONTINUE...
2439          000057          MICPC=MICPC+1
2440  014166  061620          .WORD   .SBR!.SELB!.DBRSH
2441  014170                  BB7     12$                     ;
2442          000060          MICPC=MICPC+1
2443  014170  103433          .WORD   .$.
2444  014172          14$:
2445  014172                  CALL    SCPE
2446  014172                  MOVE    @ (MICPC+3),BREG
2447          000061          MICPC=MICPC+1
2448  014172  000463          .WORD   .$.
2449  014174                  SBR     SCPE
2450          000062          MICPC=MICPC+1
2451  014174  104454          .WORD   .$.
2452  014176                  SBR     21$
2453          000063          MICPC=MICPC+1
2454  014176  100400          .WORD   .$.
2455  014200          SFLT1   OUT1,INP1,5,0
2456  014200          SXZ
2457
2458
2459                          ;*************************** TEST 2 ****************************
2460                          ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
2461                          ;* FLOAT A 1 THROUGH REGISTER OUT1 (5)
2462                          ;* FLOAT A 0THROUGH REGISTER OUT1 (5)
2463  014200          SXZ
2464                          ;;***********************************************************
2465
2466  014200          STSTN
2467                          ;  TEST 2
```

```
                                        ;---------------
014200 012737 000002 001202 TST2:       MOV     #2,$TSTNM               ; LOAD THE NO. OF THIS TEST
014206 012737 014404 001442             MOV     #TST3,NEXT              ; POINT TO THE START OF NEXT TEST.
                                ;R1 CONTAINS BASE KMC11 ADDRESS
014214 004737 035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
014220 014234                           MCT2
014222 104022                           ERROR   22                      ;TIME OUT ERROR...
014224 012706 001200                    MOV     #STACK,SP               ;RESET STACK...
014230 000177 165206                    JMP     @NEXT                   ;GO TO NEXT TEST...
014234                          MCT2:
014234                          21$:
014234                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
000000                                  MICPC=MICPC+1
014234 000400                           .WORD   .$.
014236                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
000001                                  MICPC=MICPC+1
014236 063236                           .WORD   .$.
014240                          SFLOT   OUT1,INP1,5,0,1,1$,2$,3$,4$
014240                          1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
000002                                  MICPC=MICPC+1
014240 000600                           .WORD   .$.
014242                          2$:
014242                                  MOVE    BREG,OUT1 <5>   ;SET THE BIT.
000003                                  MICPC=MICPC+1
014242 061225                           .WORD   .$.
014244                                  MOVE    INP1 <5>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
000004                                  MICPC=MICPC+1
014244 123120                           .WORD   .$.
014246                                  MOVE    BREG,SPAD <4>
000005                                  MICPC=MICPC+1
014246 063224                           .WORD   .$.
014250                                  $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...


014250                                  SUB2C   SPAD <0>,BREG,NOP
000006                                  MICPC=MICPC+1
014250 060360                           .WORD   .$.
014252                                  BZ      3$
000007                                  MICPC=MICPC+1
014252 101422                           .WORD   .$.
014254                                  MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
000010                                  MICPC=MICPC+1
014254 061224                           .WORD   .$.
014256                                  MOVE    INP1 <5>,OUT1 <CSR5>;BAD DATA...
000011                                  MICPC=MICPC+1
014256 121125                           .WORD   .$.
014260                                  MOVE    # 1,MEM                 ;TYPE OF ERROR...
000012                                  MICPC=MICPC+1
014260 002401                           .WORD   .$.
014262                                  MOVE    MEM,OUT1 <CSR3>         ;
000013                                  MICPC=MICPC+1
014262 041223                           .WORD   .$.
014264                                  MOVE    # 5,MEM         ;
000014                                  MICPC=MICPC+1
014264 002405                           .WORD   .$.
014266                                  MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
```

```
                    000015                  MICPC=MICPC+1
014266  041227                              .WORD   .$.
014270                                      CALL    ERÒR                    ;REPORT DATA ERROR.
014270                                      MOVE    # <MICPC+3>,BREG
                    000016                  MICPC=MICPC+1
014270  000420                              .WORD   .$.
014272                                      $BR     ERÒR
                    000017                  MICPC=MICPC+1
014272  104400                              .WORD   .$.
014274                                      MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                    000020                  MICPC=MICPC+1
014274  060604                              .WORD   .$.
014276                                      $BR     2$                      ;LOOP ON ERROR...
                    000021                  MICPC=MICPC+1
014276  100403                              .WORD   .$.
014300                              3$:     CALL    SCP1                    ;IS LOOP DATA SET.
014300                                      MOVE    # <MICPC+3>,BREG
                    000022                  MICPC=MICPC+1
014300  000424                              .WORD   .$.
014302                                      $BR     SCP1
                    000023                  MICPC=MICPC+1
014302  104427                              .WORD   .$.
014304                                      MOVE    SPAD <4>,BREG
                    000024                  MICPC=MICPC+1
014304  060604                              .WORD   .$.
014306                                      $BR     2$                      ;YES, DO IT.
                    000025                  MICPC=MICPC+1
014306  100403                              .WORD   .$.
014310                                      MOVE    SPAD <4>,BREG
                    000026                  MICPC=MICPC+1
014310  060604                              .WORD   .$.
014312                                      $NFBRT                          ;NO, CONTINUE...
                    000027                  MICPC=MICPC+1
014312  061620                              .WORD   .$BR!.$ELB!.DBRSH
014314                                      BR7     4$                      ;IS IT DONE?...
                    000030                  MICPC=MICPC+1
014314  103432                              .WORD   .$.
014316                                      $BR     2$                      ;NO, CONTINUE...
                    000031                  MICPC=MICPC+1
014316  100403                              .WORD   .$.
014320                              4$:
014320                              SFLOT   OUT1,INP1,5,0,0,11$,12$,13$,14$
014320                              11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
                    000032                  MICPC=MICPC+1
014320  000577                              .WORD   .$.
014322                              12$:
014322                                      MOVE    BREG,OUT1 <5>   ;SET THE BIT.
                    000033                  MICPC=MICPC+1
014322  061225                              .WORD   .$.
014324                                      MOVE    INP1 <5>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
                    000034                  MICPC=MICPC+1
014324  123120                              .WORD   .$.
014326                                      MOVE    BREG,SPAD <4>
                    000035                  MICPC=MICPC+1
014326  063224                              .WORD   .$.
014330                                      $IFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...
```

```
2580
2581
2582 014330                SUB2C   SPAD <0>,BREG,NOP
2583 000036                MICPC=MICPC+1
2584 014330 060360         .WORD   .$.
2585 014332                BZ      13$
2586 000037                MICPC=MICPC+1
2587 014332 101452         .WORD   .$.
2588 014334                MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
2589 000040                MICPC=MICPC+1
2590 014334 061224         .WORD   .$.
2591 014336                MOVE    INP1 <5>,OUT1 <CSR5>;BAD DATA...
2592 000041                MICPC=MICPC+1
2593 014336 121125         .WORD   .$.
2594 014340                MOVE    # 1,MEM                 ;TYPE OF ERROR...
2595 000042                MICPC=MICPC+1
2596 014340 002401         .WORD   .$.
2597 014342                MOVE    MEM,OUT1 <CSR3>         ;
2598 000043                MICPC=MICPC+1
2599 014342 041223         .WORD   .$.
2600 014344                MOVE    # 5,MEM         ;
2601 000044                MICPC=MICPC+1
2602 014344 002405         .WORD   .$.
2603 014346                MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
2604 000045                MICPC=MICPC+1
2605 014346 041227         .WORD   .$.
2606 014350                CALL    ERROR                   ;REPORT DATA ERROR.
2607 014350                MOVE    # <MICPC+3>,BREG
2608 000046                MICPC=MICPC+1
2609 014350 000450         .WORD   .$.
2610 014352                SBR     ERROR
2611 000047                MICPC=MICPC+1
2612 014352 104400         .WORD   .$.
2613 014354                MOVE    SPAD <4>,BREG           ;RESTORE BREG...
2614 000050                MICPC=MICPC+1
2615 014354 060604         .WORD   .$.
2616 014356                SBR     12$                     ;LOOP ON ERROR...
2617 000051                MICPC=MICPC+1
2618 014356 100433         .WORD   .$.
2619 014360          13$:  CALL    SCP1                    ;IS LOOP DATA SET.
2620 014360                MOVE    # <MICPC+3>,BREG
2621 000052                MICPC=MICPC+1
2622 014360 000454         .WORD   .$.
2623 014362                SBR     SCP1
2624 000053                MICPC=MICPC+1
2625 014362 104427         .WORD   .$.
2626 014364                MOVE    SPAD <4>,BREG
2627 000054                MICPC=MICPC+1
2628 014364 060604         .WORD   .$.
2629 014366                SBR     12$                     ;YES, DO IT.
2630 000055                MICPC=MICPC+1
2631 014366 100433         .WORD   .$.
2632 014370                MOVE    SPAD <4>,BREG
2633 000056                MICPC=MICPC+1
2634 014370 060604         .WORD   .$.
2635 014372                SHFBRT                          ;NO, CONTINUE...
```

```
                    000057                      MICPC=MICPC+1
014372  061620                                  .WORD   .SBR!.SELB!.DBRSH
014374                                          BR7     1$S             ;
                    000060                      MICPC=MICPC+1
014374  103433                                  .WORD   .$.
014376                              14$:
014376                                          CALL    SCPE
014376                                          MOVE    # <MICPC+3>,BREG
                    000061                      MICPC=MICPC+1
014376  000463                                  .WORD   .$.
014400                                          SBR     SCPE
                    000062                      MICPC=MICPC+1
014400  104454                                  .WORD   .$.
014402                                          SBR     21$
                    000063                      MICPC=MICPC+1
014402  100400                                  .WORD   .$.
014404                              $FLT1       OUT1,INP1,6,0
014404                              $XZ


                                                ;************************** TEST 3 **************************
                                                ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
                                                ;* FLOAT A 1 THROUGH REGISTER OUT1 <6>
                                                ;* FLOAT A 0THROUGH REGISTER OUT1 <6>
014404                              $XZ
                                                ;:*********************************************************************


014404                              $TSTN
                                                ;  TEST 3
                                                ;-------------
014404  012737  000003  001202      TST3:   MOV     #3,$TSTNM               ; LOAD THE NO. OF THIS TEST
014412  012737  014610  001442              MOV     #TST4,NEXT              ; POINT TO THE START OF NEXT TEST.
                                                                    ;R1 CONTAINS BASE KMC11 ADDRESS
014420  004737  035536                      JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
014424  014440                              MCT3
014426  104022                              ERROR   22                      ;TIME OUT ERROR...
014430  012706  001200                      MOV     #STACK,SP               ;RESET STACK...
014434  000177  165002                      JMP     @NEXT                   ;GO TO NEXT TEST...
014440                              MCT3:
014440                              21$:
014440                                          MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
                    000000                      MICPC=MICPC+1
014440  000400                                  .WORD   .$.
014442                                          MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
                    000001                      MICPC=MICPC+1
014442  063236                                  .WORD   .$.
014444                              $FLOT   OUT1,INP1,6,0,1,1$,2$,3$,4$
014444                              1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
                    000002                      MICPC=MICPC+1
014444  000600                                  .WORD   .$.
014446                              2$:
014446                                          MOVE    BREG,OUT1 <6>   ;SET THE BIT.
                    000003                      MICPC=MICPC+1
014446  061226                                  .WORD   .$.
014450                                          MOVE    INP1 <6>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
                    000004                      MICPC=MICPC+1
```

```
                                .WORD   .$
014450  123140                  MOVE    BREG,SPAD <4>
014452                          MICPC=MICPC+1
        000005
014452  063224                  .WORD   .$
014454                          $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...

014454                          SUB2C   SPAD <0>,BREG,NOP
        000006                  MICPC=MICPC+1
014454  060360                  .WORD   .$
014456                          BZ      3$
        000007                  MICPC=MICPC+1
014456  101422                  .WORD   .$
014460                          MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
        000010                  MICPC=MICPC+1
014460  061224                  .WORD   .$
014462                          MOVE    INP1 <6>,OUT1 <CSR5>;BAD DATA...
        000011                  MICPC=MICPC+1
014462  121145                  .WORD   .$
014464                          MOVE    # 1,MEM                 ;TYPE OF ERROR...
        000012                  MICPC=MICPC+1
014464  002401                  .WORD   .$
014466                          MOVE    MEM,OUT1 <CSR3>         ;
        000013                  MICPC=MICPC+1
014466  041223                  .WORD   .$
014470                          MOVE    # 6,MEM         ;
        000014                  MICPC=MICPC+1
014470  002406                  .WORD   .$
014472                          MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
        000015                  MICPC=MICPC+1
014472  041227                  .WORD   .$
014474                          CALL    ERROR                   ;REPORT DATA ERROR.
014474                          MOVE    # <MICPC+3>,BREG
        000016                  MICPC=MICPC+1
014474  000420                  .WORD   .$
014476                          SBR     ERROR
        000017                  MICPC=MICPC+1
014476  104400                  .WORD   .$
014500                          MOVE    SPAD <4>,BREG           ;RESTORE BREG...
        000020                  MICPC=MICPC+1
014500  060604                  .WORD   .$
014502                          SBR     2$                      ;LOOP ON ERROR...
        000021                  MICPC=MICPC+1
014502  100403                  .WORD   .$
014504                  3$:     CALL    SCP1                    ;IS LOOP DATA SET.
014504                          MOVE    # <MICPC+3>,BREG
        000022                  MICPC=MICPC+1
014504  000424                  .WORD   .$
014506                          SBR     SCP1
        000023                  MICPC=MICPC+1
014506  104427                  .WORD   .$
014510                          MOVE    SPAD <4>,BREG
        000024                  MICPC=MICPC+1
014510  060604                  .WORD   .$
014512                          SBR     2$                      ;YES, DO IT.
        000025                  MICPC=MICPC+1
```

```
014512 100403                    .WORD   $
014514                           MOVE    SPAD <4>,BREG
        000026                   MICPC=MICPC+1
014514 060604                    .WORD   .S.
014516                           SUFBRT                          ;NO, CONTINUE...
        000027                   MICPC=MICPC+1
014516 061620                    .WORD   .SBR!.SELB!.DBRSH
014520                           BBZ     4S                      ;IS IT DONE?...
        000030                   MICPC=MICPC+1
014520 103432                    .WORD   .S.
014522                           SBR     2S                      ;NO, CONTINUE...
        000031                   MICPC=MICPC+1
014522 100403                    .WORD   .S.
                        4S:
                        SFLOT   OUT1,INP1,6,0,0,11$,12$,13$,14$
                        11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
014524                          MICPC=MICPC+1
        000032                  .WORD   .S.
014524 000577           12$:
014526                          MOVE    BREG,OUT1 <6>   ;SET THE BIT.
014526                  000033  MICPC=MICPC+1
        061226                  .WORD   .S.
014530                          MOVE    INP1 <6>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
        000034                  MICPC=MICPC+1
014530 123140                   .WORD   .S.
014532                          MOVE    BREG,SPAD <4>
        000035                  MICPC=MICPC+1
014532 063224                   .WORD   .S.
014534                          SIFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...


014534
        000036          SUB2C   SPAD <0>,BREG,NOP
014534 060360           MICPC=MICPC+1
014536                  .WORD   .S.
        000037          BZ      13$
014536 101452           MICPC=MICPC+1
014540                  .WORD   .S.
        000040          MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
014540 061224           MICPC=MICPC+1
014542                  .WORD   .S.
        000041          MOVE    INP1 <6>,OUT1 <CSR5>;BAD DATA...
014542 121145           MICPC=MICPC+1
014544                  .WORD   .S.
        000042          MOVE    # 1,MEM                 ;TYPE OF ERROR...
014544 002401           MICPC=MICPC+1
014546                  .WORD   .S.
        000043          MOVE    MEM,OUT1 <CSR3>         ;
014546 041223           MICPC=MICPC+1
014550                  .WORD   .S.
        000044          MOVE    # 6,MEM         ;
014550 002406           MICPC=MICPC+1
014552                  .WORD   .S.
        000045          MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
014552 041227           MICPC=MICPC+1
014554                  .WORD   .S.
                        CALL    EROR                    ;REPORT DATA ERROR.
```

```
014554                          MOVE    # <MICPC+3>,BREG
        000046                  MICPC=MICPC+1
014554  000450                  .WORD   .$.
014556                          SBR     ERROR
        000047                  MICPC=MICPC+1
014556  104400                  .WORD   .$.
014560                          MOVE    SPAD <4>,BREG           ;RESTORE BREG...
        000050                  MICPC=MICPC+1
014560  060604                  .WORD   .$.
014562                          SBR     12$                     ;LOOP ON ERROR...
        000051                  MICPC=MICPC+1
014562  100433                  .WORD   .$.
014564                  13$:    CALL    SCP1                    ;IS LOOP DATA SET.
014564                          MOVE    # <MICPC+3>,BREG
        000052                  MICPC=MICPC+1
014564  000454                  .WORD   .$.
014566                          SBR     SCP1
        000053                  MICPC=MICPC+1
014566  104427                  .WORD   .$.
014570                          MOVE    SPAD <4>,BREG
        000054                  MICPC=MICPC+1
014570  060604                  .WORD   .$.
014572                          SBR     12$                     ;YES, DO IT.
        000055                  MICPC=MICPC+1
014572  100433                  .WORD   .$.
014574                          MOVE    SPAD <4>,BREG
        000056                  MICPC=MICPC+1
014574  060604                  .WORD   .S.
014576                          SHFRT                           ;NO, CONTINUE...
        000057                  MICPC=MICPC+1
014576  061620                  .WORD   .SBR!.SELB!.DBRSH
014600                          BR7     12$                     ;
        000060                  MICPC=MICPC+1
014600  103433                  .WORD   .S.
014602                  14$:
014602                          CALL    SCPE
014602                          MOVE    # <MICPC+3>,BREG
        000061                  MICPC=MICPC+1
014602  000463                  .WORD   .$.
014604                          SBR     SCPE
        000062                  MICPC=MICPC+1
014604  104454                  .WORD   .$.
014606                          SBR     21$
        000063                  MICPC=MICPC+1
014606  100400                  .WORD   .$.
014610                  SFLT1   OUT1,INP1,7,0
014610                  SXZ


                        ;************************** TEST 4 **************************
                        ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
                        ;* FLOAT A 1 THROUGH REGISTER OUT1 <7>
                        ;* FLOAT A 0THROUGH REGISTER OUT1 <7>
014610                  SXZ
                        ;;*********************************************************
```

```
2860  014610                          $TSTN
2861                                  ;  TEST 4
2862                                  ;--------------
2863  014610  012737  000004  001202  TST4:  MOV     #4,$TSTNM               ; LOAD THE NO. OF THIS TEST
2864  014616  012737  015014  001442          MOV     #TST5,NEXT              ; POINT TO THE START OF NEXT TEST.
2865                                                          ;R1 CONTAINS BASE KMC11 ADDRESS
2866  014624  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
2867  014630  014644                          MCT4
2868  014632  104022                          ERROR   22                      ;TIME OUT ERROR...
2869  014634  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
2870  014640  000177  164576                  JMP     @NEXT                   ;GO TO NEXT TEST...
2871  014644                          MCT4:
2872  014644                          21$:
2873  014644                                  MOVE    # 0,BREG         ;SET TO CLEAR SPAD 16
2874          000000                          MICPC=MICPC+1
2875  014646  000400                          .WORD   .$.
2876  014646                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
2877          000001                          MICPC=MICPC+1
2878  014646  063236                          .WORD   .$.
2879  014650                          $FLOT   OUT1,INP1,7,0,1,1$,2$,3$,4$
2880  014650                          1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
2881          000002                          MICPC=MICPC+1
2882  014650  000600                          .WORD   .$.
2883  014652                          2$:
2884  014652                                  MOVE    BREG,OUT1 <7>   ;SET THE BIT.
2885          000003                          MICPC=MICPC+1
2886  014652  061227                          .WORD   .$.
2887  014654                                  MOVE    INP1 <7>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
2888          000004                          MICPC=MICPC+1
2889  014654  123160                          .WORD   .$.
2890  014656                                  MOVE    BREG,SPAD <4>
2891          000005                          MICPC=MICPC+1
2892  014656  063224                          .WORD   .$.
2893  014660                                  $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...
2894
2895
2896  014660                                  SUB2C   SPAD <0>,BREG,NOP
2897          000006                          MICPC=MICPC+1
2898  014660  060360                          .WORD   .$.
2899  014662                                  BZ      3$
2900          000007                          MICPC=MICPC+1
2901  014662  101422                          .WORD   .$.
2902  014664                                  MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
2903          000010                          MICPC=MICPC+1
2904  014664  061224                          .WORD   .$.
2905  014666                                  MOVE    INP1 <7>,OUT1 <CSR5>;BAD DATA...
2906          000011                          MICPC=MICPC+1
2907  014666  121165                          .WORD   .$.
2908  014670                                  MOVE    # 1,MEM                 ;TYPE OF ERROR...
2909          000012                          MICPC=MICPC+1
2910  014670  002401                          .WORD   .$.
2911  014672                                  MOVE    MEM,OUT1 <CSR3>         ;
2912          000013                          MICPC=MICPC+1
2913  014672  041223                          .WORD   .$.
2914  014674                                  MOVE    # 7,MEM         ;
2915          000014                          MICPC=MICPC+1
```

```
2916  014674  002407                    .WORD   .$.
2917  014676                            MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
2918          000015                    MICPC=MICPC+1
2919  014676  041227                    .WORD   .$.
2920  014700                            CALL    ERROR                   ;REPORT DATA ERROR.
2921  014700                            MOVE    # <MICPC+3>,BREG
2922          000016                    MICPC=MICPC+1
2923  014700  000420                    .WORD   .$.
2924  014702                            SBR     ERROR
2925          000017                    MICPC=MICPC+1
2926  014702  104400                    .WORD   .$.
2927  014704                            MOVE    SPAD <4>,BREG           ;RESTORE BREG...
2928          000020                    MICPC=MICPC+1
2929  014704  060604                    .WORD   .$.
2930  014706                            SBR     2$                      ;LOOP ON ERROR...
2931          000021                    MICPC=MICPC+1
2932  014706  100403                    .WORD   .$.
2933  014710                    3$:     CALL    SCP1                    ;IS LOOP DATA SET.
2934  014710                            MOVE    # <MICPC+3>,BREG
2935          000022                    MICPC=MICPC+1
2936  014710  000424                    .WORD   .$.
2937  014712                            SBR     SCP1
2938          000023                    MICPC=MICPC+1
2939  014712  104427                    .WORD   .$.
2940  014714                            MOVE    SPAD <4>,BREG
2941          000024                    MICPC=MICPC+1
2942  014714  060604                    .WORD   .$.
2943  014716                            SBR     2$                      ;YES, DO IT.
2944          000025                    MICPC=MICPC+1
2945  014716  100403                    .WORD   .$.
2946  014720                            MOVE    SPAD <4>,BREG
2947          000026                    MICPC=MICPC+1
2948  014720  060604                    .WORD   .$.
2949  014722                            SHFBRT                          ;NO, CONTINUE...
2950          000027                    MICPC=MICPC+1
2951  014722  061620                    .WORD   .SBR!.SELB!.DBRSH
2952  014724                            BB7     4$                      ;IS IT DONE?...
2953          000030                    MICPC=MICPC+1
2954  014724  103432                    .WORD   .$.
2955  014726                            SBR     2$                      ;NO, CONTINUE...
2956          000031                    MICPC=MICPC+1
2957  014726  100403                    .WORD   .$.
2958  014730                    4$:
2959  014730                    SFLOT   OUT1,INP1,7,0,0,11$,12$,13$,14$
2960  014730                    11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
2961          000032                    MICPC=MICPC+1
2962  014730  000577                    .WORD   .$.
2963  014732                    12$:
2964  014732                            MOVE    BREG,OUT1 <7>   ;SET THE BIT.
2965          000033                    MICPC=MICPC+1
2966  014732  051227                    .WORD   .$.
2967  014734                            MOVE    INP1 <7>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
2968          000034                    MICPC=MICPC+1
2969  014734  123160                    .WORD   .$.
2970  014736                            MOVE    BREG,SPAD <4>
2971          000035                    MICPC=MICPC+1
```

```
014736 063224          .WORD   .$.
014740                 SIFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...


014740                 SUB2C   SPAD <0>,BREG,NOP
       000036          MICPC=MICPC+1
014740 060360          .WORD   .$.
014742                 BZ      13$
       000037          MICPC=MICPC+1
014742 101452          .WORD   .$.
014744                 MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
       000040          MICPC=MICPC+1
014744 061224          .WORD   .$.
014746                 MOVE    INP1 <7>,OUT1 <CSR5>;BAD DATA...
       000041          MICPC=MICPC+1
014746 121165          .WORD   .$.
014750                 MOVE    #1,MEM                  ;TYPE OF ERROR...
       000042          MICPC=MICPC+1
014750 002401          .WORD   .$.
014752                 MOVE    MEM,OUT1 <CSR3>         ;
       000043          MICPC=MICPC+1
014752 041223          .WORD   .$.
014754                 MOVE    #7,MEM          ;
       000044          MICPC=MICPC+1
014754 002407          .WORD   .$.
014756                 MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
       000045          MICPC=MICPC+1
014756 041227          .WORD   .$.
014760                 CALL    ERROR                   ;REPORT DATA ERROR.
014760                 MOVE    # <MICPC+3>,BREG
       000046          MICPC=MICPC+1
014760 000450          .WORD   .$.
014762                 SBR     ERROR
       000047          MICPC=MICPC+1
014762 104400          .WORD   .$.
014764                 MOVE    SPAD <4>,BREG           ;RESTORE BREG...
       000050          MICPC=MICPC+1
014764 060604          .WORD   .$.
014766                 SBR     12$                     ;LOOP ON ERROR...
       000051          MICPC=MICPC+1
014766 100433          .WORD   .$.
014770          13$:   CALL    SCP1                    ;IS LOOP DATA SET.
014770                 MOVE    # <MICPC+3>,BREG
       000052          MICPC=MICPC+1
014770 000454          .WORD   .$.
014772                 SBR     SCP1
       000053          MICPC=MICPC+1
014772 104427          .WORD   .$.
014774                 MOVE    SPAD <4>,BREG
       000054          MICPC=MICPC+1
014774 060604          .WORD   .$.
014776                 SBR     12$                     ;YES, DO IT.
       000055          MICPC=MICPC+1
014776 100433          .WORD   .$.
015000                 MOVE    SPAD <4>,BREG
       000056          MICPC=MICPC+1
```

```
015000  060604                  .WORD   .$.
015002                          $HFBRT                          ;NO, CONTINUE...
        000057                  MICPC=MICPC+1
015002  061620                  .WORD   .SBR!.SELB!.DBRSH
015004                          B87     12$                     ;
        000060                  MICPC=MICPC+1
015004  103433                  .WORD   .$.
015006                  14$:
015006                          CALL    SCPE
015006                          MOVE    # <MICPC+3>,BREG
        000061                  MICPC=MICPC+1
015006  000463                  .WORD   .$.
015010                          $BR     SCPE
        000062                  MICPC=MICPC+1
015010  104454                  .WORD   .$.
015012                          $BR     21$
        000063                  MICPC=MICPC+1
015012  100400                  .WORD   .$.
015014                  $FLT1   OUT1,INP1,10,236
015014                  $XZ


                                ;*************************** TEST 5 ****************************
                                ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
                                ;* FLOAT A 1 THROUGH REGISTER OUT1 <10>
                                ;* FLOAT A 0THROUGH REGISTER OUT1 <10>
                                ;* THE NPR RQ BIT (BIT0) IS MASKED DURING THIS TEST.
015014                  $XZ
                                ;:***************************************************************
015014                  $TSTN
                                ;  TEST 5
                                ;-------------
015014  012737  000005  001202  TST5:  MOV     #5,$TSTNM               ; LOAD THE NO. OF THIS TEST
015022  012737  015234  001442         MOV     #TST6,NEXT              ; POINT TO THE START OF NEXT TEST.
                                                              ;R1 CONTAINS BASE KMC11 ADDRESS
015030  004737  035536                 JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
015034  015050                         MCT5
015036  104022                         ERROR   22                      ;TIME OUT ERROR...
015040  012706  001200                 MOV     #STACK,SP               ;RESET STACK...
015044  000177  164372                 JMP     @NEXT                   ;GO TO NEXT TEST...
015050                  MCT5:
015050                  21$:
015050                          MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
        000000                  MICPC=MICPC+1
015050  000400                  .WORD   .$.
015052                          MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
        000001                  MICPC=MICPC+1
015052  063236                  .WORD   .$.
015054                  SFLOT   OUT1,INP1,10,236,1,1$,2$,3$,4$
015054                  1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
        000002                  MICPC=MICPC+1
015054  000600                  .WORD   .$.
015056                  2$:
015056                          MOVE    BREG,SPAD <4>           ;SAVE BREGISTER.
        000003                  MICPC=MICPC+1
```

```
3084  015056  063224                .WORD   .$.
3085  015060                        MOVE    # 236,MEM               ;CLEAR THE UNWANTED BITS
3086          000004                MICPC=MICPC+1
3087  015060  002636                .WORD   .$.
3088  015062                        MOVE    MEM,SPAD <1>            ;CLEAR THE UNWANTED BITS.
3089          000005                MICPC=MICPC+1
3090  015062  043221                .WORD   .$.
3091  015064                        AND     SPAD <1>,BREG,BREG      ;CLEAR THE UNWANTED BITS.
3092          000006                MICPC=MICPC+1
3093  015064  060661                .WORD   .$.
3094  015066                        MOVE    BREG,OUT1 <10>  ;SET THE BIT.
3095          000007                MICPC=MICPC+1
3096  015066  061230                .WORD   .$.
3097  015070                        MOVE    INP1 <10>,SPAD <0>      ;GET THE "FOUND" IN SCRATCH PAD.
3098          000010                MICPC=MICPC+1
3099  015070  123200                .WORD   .$.
3100  015072                        $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...
3101
3102
3103  015072                        SUB2C   SPAD <0>,BREG,NOP
3104          000011                MICPC=MICPC+1
3105  015072  060360                .WORD   .$.
3106  015074                        BZ      3$
3107          000012                MICPC=MICPC+1
3108  015074  101425                .WORD   .$.
3109  015076                        MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
3110          000013                MICPC=MICPC+1
3111  015076  061224                .WORD   .$.
3112  015100                        MOVE    INP1 <10>,OUT1 <CSR5>;BAD DATA...
3113          000014                MICPC=MICPC+1
3114  015100  121205                .WORD   .$.
3115  015102                        MOVE    # 1,MEM                 ;TYPE OF ERROR...
3116          000015                MICPC=MICPC+1
3117  015102  002401                .WORD   .$.
3118  015104                        MOVE    MEM,OUT1 <CSR3>         ;
3119          000016                MICPC=MICPC+1
3120  015104  041223                .WORD   .$.
3121  015106                        MOVE    # 10,MEM                ;
3122          000017                MICPC=MICPC+1
3123  015106  002410                .WORD   .$.
3124  015110                        MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
3125          000020                MICPC=MICPC+1
3126  015110  041227                .WORD   .$.
3127  015112                        CALL    ERROR                   ;REPORT DATA ERROR.
3128  015112                        MOVE    # <MICPC+3>,BREG
3129          000021                MICPC=MICPC+1
3130  015112  000423                .WORD   .$.
3131  015114                        SBR     ERROR
3132          000022                MICPC=MICPC+1
3133  015114  104400                .WORD   .$.
3134  015116                        MOVE    SPAD <4>,BREG           ;RESTORE BREG...
3135          000023                MICPC=MICPC+1
3136  015116  060604                .WORD   .$.
3137  015120                        SBR     2$                      ;LOOP ON ERROR...
3138          000024                MICPC=MICPC+1
3139  015120  100403                .WORD   .$.
```

```
3140  015122                  3$:     CALL    SCP1                    ;IS LOOP DATA SET.
3141  015122                          MOVE    # <MICPC+3>,BREG
3142          000025                  MICPC=MICPC+1
3143  015122  000427                  .WORD   .$.
3144  015124                          SBR     SCP1
3145          000026                  MICPC=MICPC+1
3146  015124  104427                  .WORD   .$.
3147  015126                          MOVE    SPAD <4>,BREG
3148          000027                  MICPC=MICPC+1
3149  015126  060604                  .WORD   .$.
3150  015130                          SBR     2$                      ;YES, DO IT.
3151          000030                  MICPC=MICPC+1
3152  015130  100403                  .WORD   .$.
3153  015132                          MOVE    SPAD <4>,BREG
3154          000031                  MICPC=MICPC+1
3155  015132  060604                  .WORD   .$.
3156  015134                          SHFBRT                          ;NO, CONTINUE...
3157          000032                  MICPC=MICPC+1
3158  015134  061620                  .WORD   .SBR!.SELB!.DBRSH
3159  015136                          BB7     4$                      ;IS IT DONE?...
3160          000033                  MICPC=MICPC+1
3161  015136  103435                  .WORD   .$.
3162  015140                          SBR     2$                      ;NO, CONTINUE...
3163          000034                  MICPC=MICPC+1
3164  015140  100403                  .WORD   .$.
3165  015142          4$:
3166  015142          SFLOT   OUT1,INP1,10,236,0,11$,12$,13$,14$
3167  015142          11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
3168          000035          MICPC=MICPC+1
3169  015142  000577          .WORD   .$.
3170  015144          12$:
3171  015144                  MOVE    BREG,SPAD <4>           ;SAVE BREGISTER.
3172          000036          MICPC=MICPC+1
3173  015144  063224          .WORD   .$.
3174  015146                  MOVE    # 236,MEM               ;CLEAR THE UNWANTED BITS
3175          000037          MICPC=MICPC+1
3176  015146  002636          .WORD   .$.
3177  015150                  MOVE    MEM,SPAD <1>            ;CLEAR THE UNWANTED BITS.
3178          000040          MICPC=MICPC+1
3179  015150  043221          .WORD   .$.
3180  015152                  AND     SPAD <1>,BREG,BREG      ;CLEAR THE UNWANTED BITS.
3181          000041          MICPC=MICPC+1
3182  015152  060661          .WORD   .$.
3183  015154                  MOVE    BREG,OUT1 <10>  ;SET THE BIT.
3184          000042          MICPC=MICPC+1
3185  015154  061230          .WORD   .$.
3186  015156                  MOVE    INP1 <10>,SPAD <0>      ;GET THE "FOUND" IN SCRATCH PAD.
3187          000043          MICPC=MICPC+1
3188  015156  123200          .WORD   .$.
3189  015160                  SIFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...
3190
3191
3192  015160                  SUB2C   SPAD <0>,BREG,NOP
3193          000044          MICPC=MICPC+1
3194  015160  060360          .WORD   .$.
3195  015162                  BZ      13$
```

```
3196                000045                  MICPC=MICPC+1
3197  015162        101460                  .WORD   .$.
3198  015164                                MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
3199                000046                  MICPC=MICPC+1
3200  015164        061224                  .WORD   .$.
3201  015166                                MOVE    INP1 <10>,OUT1 <CSR5>;BAD DATA...
3202                000047                  MICPC=MICPC+1
3203  015166        121205                  .WORD   .$.
3204  015170                                MOVE    # 1,MEM                 ;TYPE OF ERROR...
3205                000050                  MICPC=MICPC+1
3206  015170        002401                  .WORD   .$.
3207  015172                                MOVE    MEM,OUT1 <CSR3>         ;
3208                000051                  MICPC=MICPC+1
3209  015172        041223                  .WORD   .$.
3210  015174                                MOVE    # 10,MEM                ;
3211                000052                  MICPC=MICPC+1
3212  015174        002410                  .WORD   .$.
3213  015176                                MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
3214                000053                  MICPC=MICPC+1
3215  015176        041227                  .WORD   .$.
3216  015200                                CALL    EROR                    ;REPORT DATA ERROR.
3217  015200                                MOVE    # <MICPC+3>,BREG
3218                000054                  MICPC=MICPC+1
3219  015200        000456                  .WORD   .$.
3220  015202                                SBR     EROR
3221                000055                  MICPC=MICPC+1
3222  015202        104400                  .WORD   .$.
3223  015204                                MOVE    SPAD <4>,BREG           ;RESTORE BREG...
3224                000056                  MICPC=MICPC+1
3225  015204        060604                  .WORD   .$.
3226  015206                                SBR     12$                     ;LOOP ON ERROR...
3227                000057                  MICPC=MICPC+1
3228  015206        100436                  .WORD   .$.
3229  015210                        13$:    CALL    SCP1                    ;IS LOOP DATA SET.
3230  015210                                MOVE    # <MICPC+3>,BREG
3231                000060                  MICPC=MICPC+1
3232  015210        000462                  .WORD   .$.
3233  015212                                SBR     SCP1
3234                000061                  MICPC=MICPC+1
3235  015212        104427                  .WORD   .$.
3236  015214                                MOVE    SPAD <4>,BREG
3237                000062                  MICPC=MICPC+1
3238  015214        060604                  .WORD   .$.
3239  015216                                SBR     12$                     ;YES, DO IT.
3240                000063                  MICPC=MICPC+1
3241  015216        100436                  .WORD   .$.
3242  015220                                MOVE    SPAD <4>,BREG
3243                000064                  MICPC=MICPC+1
3244  015220        060604                  .WORD   .$.
3245  015222                                SHFBRT                          ;NO, CONTINUE...
3246                000065                  MICPC=MICPC+1
3247  015222        061620                  .WORD   .SBR!.SELB!.DBRSH
3248  015224                                BBT     12$                     ;
3249                000066                  MICPC=MICPC+1
3250  015224        103436                  .WORD   .$.
3251  015226                        14$:
```

```
                                        CALL    SCPE
                                        MOVE    @(MICPC+3),BREG
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        $BR     SCPE
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        $BR     21$
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                SFLT1   OUT1,INPI,I1,015
                                        $XZ


                                        ;*************************** TEST 6 ****************************
                                        ;* MICRO PROCESSOR OUT1 REGISTER WRITE/READ TEST.
                                        ;* FLOAT A 1 THROUGH REGISTER OUT1 (I1)
                                        ;* FLOAT A 0THROUGH REGISTER OUT1 (I1)
                                        ;* THE BR RQ BIT, PGM CLOCK BIT, FORCE POWER FAIL BIT
                                        ;* (BITS 7,4,1) ARE ALL MASKED DURING THIS TEST
                                SXZ
                                        ;;***********************************************************

                                STSTN

                                        ;  TEST 6
                                        ;-------------
001202  TST6:   MOV     #6,STSTNM               ; LOAD THE NO. OF THIS TEST
001442          MOV     #TST7,NEXT              ; POINT TO THE START OF NEXT TEST.
                                        ;R1 CONTAINS BASE KMC11 ADDRESS
035536          JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
                MCT6
                ERROR   22                      ;TIME OUT ERROR...
001200          MOV     #STACK,SP               ;RESET STACK...
164152          JMP     @NEXT                   ;GO TO NEXT TEST...
                MCT6:
                21$:
                        MOVE    # 0,BREG         ;SET TO CLEAR SPAD 16
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
                        MICPC=MICPC+1
                        .WORD   .$.
                SFLOT   OUT1,INPI,I1,015,1,1$,2$,3$,4$
                1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
                        MICPC=MICPC+1
                        .WORD   .$.
                2$:
                        MOVE    BREG,SPAD (4)           ;SAVE BREGISTER.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    # 015,MEM               ;CLEAR THE UNWANTED BITS
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    MEM,SPAD (1)            ;CLEAR THE UNWANTED BITS.
                        MICPC=MICPC+1
                        .WORD   .$.
```

```
015304                  AND     SPAD (1),BREG,BREG      ;CLEAR THE UNWANTED BITS.
        000006          MICPC=MICPC+1
015304  060661          .WORD   .$.
015306                  MOVE    BREG,OUT1 (11)  ;SET THE BIT.
        000007          MICPC=MICPC+1
015306  061231          .WORD   .$.
015310                  MOVE    INP1 (11),SPAD (0)      ;GET THE "FOUND" IN SCRATCH PAD.
        000010          MICPC=MICPC+1
015310  123220          .WORD   .$.
015312                  MOVE    # 20,MEM                ;SET TO SET PGM.CLK BIT.
        000011          MICPC=MICPC+1
015312  002420          .WORD   .$.
015314                  MOVE    MEM,SPAD  (1)           ; " " " " " " .
        000012          MICPC=MICPC+1
015314  043221          .WORD   .$.
015316                  OR      SPAD (1),BREG,BREG      ;SET PGM.CLK. BIT IN BREG.
        000013          MICPC=MICPC+1
015316  060701          .WORD   .$.
015320                  SIFEQ   BREG,SPAD (0)  3$       ;CHECK THE DATA...

015320                  SUB2C   SPAD (0),BREG,NOP
        000014          MICPC=MICPC+1
015320  060360          .WORD   .$.
015322                  BZ      3$
        000015          MICPC=MICPC+1
015322  101430          .WORD   .$.
015324                  MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
        000016          MICPC=MICPC+1
015324  061224          .WORD   .$.
015326                  MOVE    INP1 (11),OUT1 (CSR5);BAD DATA...
        000017          MICPC=MICPC+1
015326  121225          .WORD   .$.
015330                  MOVE    # 1,MEM                 ;TYPE OF ERROR...
        000020          MICPC=MICPC+1
015330  002401          .WORD   .$.
015332                  MOVE    MEM,OUT1 (CSR3)         ;
        000021          MICPC=MICPC+1
015332  041223          .WORD   .$.
015334                  MOVE    # 11,MEM                ;
        000022          MICPC=MICPC+1
015334  002411          .WORD   .$.
015336                  MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
        000023          MICPC=MICPC+1
015336  041227          .WORD   .$.
015340                  CALL    EROR                    ;REPORT DATA ERROR.
015340                  MOVE    # (MICPC+3),BREG
        000024          MICPC=MICPC+1
015340  000426          .WORD   .$.
015342                  SBR     EROR
        000025          MICPC=MICPC+1
015342  104400          .WORD   .$.
015344                  MOVE    SPAD (4),BREG           ;RESTORE BREG...
        000026          MICPC=MICPC+1
015344  060604          .WORD   .$.
015346                  SBR     2S                      ;LOOP ON ERROR...
```

```
3364                    000027
3365  015346  100403
3366  015350
3367  015350
3368                    000030
3369  015350  000432
3370  015352
3371                    000031
3372  015352  104427
3373  015354
3374                    000032
3375  015354  060604
3376  015356
3377                    000033
3378  015356  100403
3379  015360
3380                    000034
3381  015360  060604
3382  015362
3383                    000035
3384  015362  061620
3385  015364
3386                    000036
3387  015364  103440
3388  015366
3389                    000037
3390  015366  100403
3391  015370
3392  015370
3393  015370
3394                    000040
3395  015370  000577
3396  015372
3397  015372
3398                    000041
3399  015372  063224
3400  015374
3401                    000042
3402  015374  002415
3403  015376
3404                    000043
3405  015376  043221
3406  015400
3407                    000044
3408  015400  060661
3409  015402
3410                    000045
3411  015402  061231
3412  015404
3413                    000046
3414  015404  123220
3415  015406
3416                    000047
3417  015406  002420
3418  015410
3419                    000050
```

```
                MICPC=MICPC+1
                .WORD   .$.
3$:             CALL    SCP1                    ;IS LOOP DATA SET.
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;YES, DO IT.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SHFBRT                          ;NO, CONTINUE...
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                BR7     4$                      ;IS IT DONE?...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;NO, CONTINUE...
                MICPC=MICPC+1
                .WORD   .$.
4$:
SFLOT   OUT1,INP1,11,015,0,11$,12$,13$,14$
11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
                MICPC=MICPC+1
                .WORD   .$.
12$:
                MOVE    BREG,SPAD <4>           ;SAVE BREGISTER.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 015,MEM               ;CLEAR THE UNWANTED BITS
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,SPAD <1>            ;CLEAR THE UNWANTED BITS.
                MICPC=MICPC+1
                .WORD   .$.
                AND     SPAD <1>,BREG,BREG      ;CLEAR THE UNWANTED BITS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 <11>  ;SET THE BIT.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    INP1 <11>,SPAD <0>      ;GET THE "FOUND" IN SCRATCH PAD.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 20,MEM                ;SET TO SET PGM.CLK BIT.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,SPAD  <1>           ; " " " " " " .
                MICPC=MICPC+1
```

```
3420  015410  043221                    .WORD   .$.
3421  015412                            OR      SPAD <1>,BREG,BREG      ;SET PGM.CLK. BIT IN BREG.
3422          000051                    MICPC=MICPC+1
3423  015412  060701                    .WORD   .$.
3424  015414                            SIFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...
3425
3426
3427  015414                            SUB2C   SPAD <0>,BREG,NOP
3428          000052                    MICPC=MICPC+1
3429  015414  060360                    .WORD   .$.
3430  015416                            BZ      13$
3431          000053                    MICPC=MICPC+1
3432  015416  101466                    .WORD   .$.
3433  015420                            MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
3434          000054                    MICPC=MICPC+1
3435  015420  061224                    .WORD   .$.
3436  015422                            MOVE    INP1 <11>,OUT1 <CSR5>;BAD DATA...
3437          000055                    MICPC=MICPC+1
3438  015422  121225                    .WORD   .$.
3439  015424                            MOVE    # 1,MEM                 ;TYPE OF ERROR...
3440          000056                    MICPC=MICPC+1
3441  015424  002401                    .WORD   .$.
3442  015426                            MOVE    MEM,OUT1 <CSR3>         ;
3443          000057                    MICPC=MICPC+1
3444  015426  041223                    .WORD   .$.
3445  015430                            MOVE    # 11,MEM                ;
3446          000060                    MICPC=MICPC+1
3447  015430  002411                    .WORD   .$.
3448  015432                            MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
3449          000061                    MICPC=MICPC+1
3450  015432  041227                    .WORD   .$.
3451  015434                            CALL    ERROR                   ;REPORT DATA ERROR.
3452  015434                            MOVE    # <MICPC+3>,BREG
3453          000062                    MICPC=MICPC+1
3454  015434  000464                    .WORD   .$.
3455  015436                            SBR     ERROR
3456          000063                    MICPC=MICPC+1
3457  015436  104400                    .WORD   .$.
3458  015440                            MOVE    SPAD <4>,BREG           ;RESTORE BREG...
3459          000064                    MICPC=MICPC+1
3460  015440  060604                    .WORD   .$.
3461  015442                            SBR     12$                     ;LOOP ON ERROR...
3462          000065                    MICPC=MICPC+1
3463  015442  100441                    .WORD   .$.
3464  015444                    13$:    CALL    SCP1                    ;IS LOOP DATA SET.
3465  015444                            MOVE    # <MICPC+3>,BREG
3466          000066                    MICPC=MICPC+1
3467  015444  000470                    .WORD   .$.
3468  015446                            SBR     SCP1
3469          000067                    MICPC=MICPC+1
3470  015446  104427                    .WORD   .$.
3471  015450                            MOVE    SPAD <4>,BREG
3472          000070                    MICPC=MICPC+1
3473  015450  060604                    .WORD   .$.
3474  015452                            SBR     12$                     ;YES, DO IT.
3475          000071                    MICPC=MICPC+1
```

```
3476  015462  100441                        .WORD   .$
3477  015464                                MOVE    SPAD <4>,BREG
3478          000072                        MICPC=MICPC+1
3479  015466  060604                        .WORD   .$.
3480  015470                                SWFBRT                          ;NO, CONTINUE...
3481          000073                        MICPC=MICPC+1
3482  015472  061620                        .WORD   .SBR!.SELB!.DBRSH
3483  015474                                BR7     1$                      ;
3484          000074                        MICPC=MICPC+1
3485  015476  103441                        .WORD   .$.
3486  015500                        14$:
3487  015502                                CALL    SCPE
3488  015504                                MOVE    # <MICPC+3>,BREG
3489          000075                        MICPC=MICPC+1
3490  015506  000477                        .WORD   .$.
3491  015510                                SBR     SCPE
3492          000076                        MICPC=MICPC+1
3493  015512  104454                        .WORD   .$.
3494  015514                                SBR     21$
3495          000077                        MICPC=MICPC+1
3496  015516  100400                        .WORD   .$.
3497  015520                        SFLT1   OUTO,INPO,0,0
3498  015520                        SXZ

3499
3500
3501                                        ;************************** TEST 7 **************************
3502                                        ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
3503                                        ;* FLOAT A 1 THROUGH REGISTER OUTO <0>
3504                                        ;* FLOAT A 0THROUGH REGISTER OUTO <0>
3505  015520                        SXZ
3506                                        ;;*********************************************************************
3507
3508  015520                        $TSTN
3509                                        ;  TEST 7
3510                                        ;-------------
3511  015520  012737  000007  001202  TST7:  MOV     #7,$TSTNM               ; LOAD THE NO. OF THIS TEST
3512  015526  012737  015674  001442        MOV     #TST10,NEXT             ; POINT TO THE START OF NEXT TEST.
3513                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
3514  015534  004737  035536                JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
3515  015540  015624                        MCT7
3516  015542  104022                        ERROR   22                      ;TIME OUT ERROR...
3517  015544  012706  001200                MOV     #STACK,SP               ;RESET STACK...
3518  015550  000177  163716                JMP     @NEXT                   ;GO TO NEXT TEST...
3519  015554                        MCT7:
3520  015554                        21$:
3521  015554                                MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
3522          000000                        MICPC=MICPC+1
3523  015560  000400                        .WORD   .$.
3524  015562                                MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
3525          000001                        MICPC=MICPC+1
3526  015564  063236                        .WORD   .$.
3527  015566                        $FLOT   OUTO,INPO,0,0,1,1$,2$,3$,4$
3528  015566                        1$:  MOVE   # 200,BREG  ;START WITH BIT 7.
3529          000002                        MICPC=MICPC+1
3530  015566  000600                        .WORD   .$.
3531  015570                        2$:
```

```
3532 015532                    MOVE    BREG,OUT0 <0>   ;SET THE BIT.
3533                           MICPC=MICPC+1
3534 015532 000003             .WORD   .$.
3535 015534 062220             MOVE    INPO <0>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
3536                           MICPC=MICPC+1
3537 015534 000004             .WORD   .$.
3540 015536 023000             MOVE    BREG,SPAD <4>
3541                           MICPC=MICPC+1
3542 015536 000005             .WORD   .$.
3543 015540 063224             $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...
3544 015540
3545                           SUB2C   SPAD <0>,BREG,NOP
3546 015540 000006             MICPC=MICPC+1
3547 015542 060360             .WORD   .$.
3550                           BZ      3$
3551 015542 000007             MICPC=MICPC+1
3552 015544 101422             .WORD   .$.
3553                           MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
3554 015544 000010             MICPC=MICPC+1
3555 015546 061224             .WORD   .$.
3556                           MOVE    INPO <0>,OUT1 <CSR5>;BAD DATA...
3557 015546 000011             MICPC=MICPC+1
3560 015550 021005             .WORD   .$.
3561                           MOVE    #1,MEM                  ;TYPE OF ERROR...
3562 015550 000012             MICPC=MICPC+1
3563 015552 002401             .WORD   .$.
3564                           MOVE    MEM,OUT1 <CSR3>         ;
3565 015552 000013             MICPC=MICPC+1
3566 015554 041223             .WORD   .$.
3567                           MOVE    #0,MEM          ;
3570 015554 000014             MICPC=MICPC+1
3571 015556 002400             .WORD   .$.
3572                           MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
3573 015556 000015             MICPC=MICPC+1
3574 015560 041227             .WORD   .$.
3575                           CALL    ERROR                   ;REPORT DATA ERROR.
3576 015560                    MOVE    # <MICPC+3>,BREG
3577                           MICPC=MICPC+1
3600 015560 000016             .WORD   .$.
3601 015562 000420             SBR     ERROR
3602                           MICPC=MICPC+1
3603 015562 000017             .WORD   .$.
3604 015564 104400             MOVE    SPAD <4>,BREG           ;RESTORE BREG...
3605                           MICPC=MICPC+1
3606 015564 000020             .WORD   .$.
3607 015566 060604             SBR     2$                      ;LOOP ON ERROR...
3610                           MICPC=MICPC+1
3611 015566 000021             .WORD   .$.
3612 015570 100403     3$:     CALL    SCP1                    ;IS LOOP DATA SET.
3613                           MOVE    # <MICPC+3>,BREG
3614 015570 000022             MICPC=MICPC+1
3615 015572 000424             .WORD   .$.
3616                           SBR     SCP1
3617 015572 000023             MICPC=MICPC+1
3620 015572 104427             .WORD   .$.
```

```
015574                  MOVE    SPAD <4>,BREG
        000024          MICPC=MICPC+1
015574  060604          .WORD   .S.
015576                  SBR     2$                      ;YES, DO IT.
        000025          MICPC=MICPC+1
015576  100403          .WORD   .S.
015600                  MOVE    SPAD <4>,BREG
        000026          MICPC=MICPC+1
015600  060604          .WORD   .S.
015602                  SHFRT                           ;NO, CONTINUE...
        000027          MICPC=MICPC+1
015602  061620          .WORD   .SBR!.SELB!.DBRSH
015604                  BR7     4$                      ;IS IT DONE?...
        000030          MICPC=MICPC+1
015604  103432          .WORD   .S.
015606                  SBR     2$                      ;NO, CONTINUE...
        000031          MICPC=MICPC+1
015606  100403          .WORD   .S.
015610          4$:
015610          SFLOT   OUT0,INP0,0,0,0,11$,12$,13$,14$
015610          11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
        000032          MICPC=MICPC+1
015610  000577          .WORD   .S.
015612          12$:
015612                  MOVE    BREG,OUT0 <0>   ;SET THE BIT.
        000033          MICPC=MICPC+1
015612  062220          .WORD   .S.
015614                  MOVE    INP0 <0>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
        000034          MICPC=MICPC+1
015614  023000          .WORD   .S.
015616                  MOVE    BREG,SPAD <4>
        000035          MICPC=MICPC+1
015616  063224          .WORD   .S.
015620                  SIFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...

015620                  SUB2C   SPAD <0>,BREG,NOP
        000036          MICPC=MICPC+1
015620  060360          .WORD   .S.
015622                  BZ      13$
        000037          MICPC=MICPC+1
015622  101452          .WORD   .S.
015624                  MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
        000040          MICPC=MICPC+1
015624  061224          .WORD   .S.
015626                  MOVE    INP0 <0>,OUT1 <CSR5>;BAD DATA...
        000041          MICPC=MICPC+1
015626  021005          .WORD   .S.
015630                  MOVE    # 1,MEM                 ;TYPE OF ERROR...
        000042          MICPC=MICPC+1
015630  002401          .WORD   .S.
015632                  MOVE    MEM,OUT1 <CSR3>         ;
        000043          MICPC=MICPC+1
015632  041223          .WORD   .S.
015634                  MOVE    # 0,MEM         ;
        000044          MICPC=MICPC+1
```

```
015634 002400                  .WORD   .$.
015636                         MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
000045                         MICPC=MICPC+1
015636 041227                  .WORD   .$.
015640                         CALL    ERROR                   ;REPORT DATA ERROR.
                               MOVE    # <MICPC+3>,BREG
000046                         MICPC=MICPC+1
015640 000450                  .WORD   .$.
015642                         SBR     ERROR
000047                         MICPC=MICPC+1
015642 104400                  .WORD   .$.
015644                         MOVE    SPAD <4>,BREG           ;RESTORE BREG...
000050                         MICPC=MICPC+1
015644 060604                  .WORD   .$.
015646                         SBR     13$                     ;LOOP ON ERROR...
000051                         MICPC=MICPC+1
015646 100433                  .WORD   .$.
015650                 13$:    CALL    SCP1                    ;IS LOOP DATA SET.
                               MOVE    # <MICPC+3>,BREG
000052                         MICPC=MICPC+1
015650 000454                  .WORD   .$.
015652                         SBR     SCP1
000053                         MICPC=MICPC+1
015652 104427                  .WORD   .$.
015654                         MOVE    SPAD <4>,BREG
000054                         MICPC=MICPC+1
015654 060604                  .WORD   .$.
015656                         SBR     14$                     ;YES, DO IT.
000055                         MICPC=MICPC+1
015656 100433                  .WORD   .$.
015660                         MOVE    SPAD <4>,BREG
000056                         MICPC=MICPC+1
015660 060604                  .WORD   .$.
015662                         SHFRT                           ;NO, CONTINUE...
000057                         MICPC=MICPC+1
015662 061620                  .WORD   .SBR!.SELB!.DBRSH
015664                         BR7     12$                     ;
000060                         MICPC=MICPC+1
015664 103433                  .WORD   .$.
015666                 14$:
015666                         CALL    SCPE
                               MOVE    # <MICPC+3>,BREG
000061                         MICPC=MICPC+1
015666 000463                  .WORD   .$.
015670                         SBR     SCPE
000062                         MICPC=MICPC+1
015670 104454                  .WORD   .$.
015672                         SBR     21$
000063                         MICPC=MICPC+1
015672 100400                  .WORD   .$.
015674                 $FLT1   OUT0,INP0,1,0
015674                 $XZ
```

```
;************************** TEST 10 **************************
;* MICRO PROCESSOR OUT0 REGISTER WRITE/READ TEST.
```

```
                                        ;* FLOAT A 1 THROUGH REGISTER OUT0 <1>
                                        ;* FLOAT A 0THROUGH REGISTER OUT0 <1>
015674                          SXZ
                                        ;;*****************************************************************
015674                          $TSTN
                                        ;  TEST 10
                                        ;-------------
015674  012737  000010  001202  TST10:  MOV     #10,$TSTNM              ; LOAD THE NO. OF THIS TEST
015702  012737  016100  001442          MOV     #TST11,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
015710  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
015714  015730                          MCT10
015716  104022                          ERROR   22                      ;TIME OUT ERROR...
015720  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
015724  000177  163512                  JMP     @NEXT                   ;GO TO NEXT TEST...
015730                          MCT10:
015730                          21$:
015730                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
        000000                          MICPC=MICPC+1
015730  000400                          .WORD   .$.
015732                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
        000001                          MICPC=MICPC+1
015732  063236                          .WORD   .$.
015734                          SFLOT   OUT0,INP0,1,0,1,1$,2$,3$,4$
015734                          1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
        000002                          MICPC=MICPC+1
015734  000600                          .WORD   .$.
015736                          2$:
015736                                  MOVE    BREG,OUT0 <1>   ;SET THE BIT.
        000003                          MICPC=MICPC+1
015736  062221                          .WORD   .$.
015740                                  MOVE    INP0 <1>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
        000004                          MICPC=MICPC+1
015740  023020                          .WORD   .$.
015742                                  MOVE    BREG,SPAD <4>
        000005                          MICPC=MICPC+1
015742  063224                          .WORD   .$.
015744                                  $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...


015744                                  SUB2C   SPAD <0>,BREG,NOP
        000006                          MICPC=MICPC+1
015744  060360                          .WORD   .$.
015746                                  BZ      3$
        000007                          MICPC=MICPC+1
015746  101422                          .WORD   .$.
015750                                  MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
        000010                          MICPC=MICPC+1
015750  061224                          .WORD   .$.
015752                                  MOVE    INP0 <1>,OUT1 <CSR5>;BAD DATA...
        000011                          MICPC=MICPC+1
015752  021025                          .WORD   .$.
015754                                  MOVE    # 1,MEM                 ;TYPE OF ERROR...
        000012                          MICPC=MICPC+1
015754  002401                          .WORD   .$.
```

```
3756  015756                    MOVE    MEM,OUT1 <CSR3>         ;
3757          000013            MICPC=MICPC+1
3758  015756  041223            .WORD   .$.
3759  015760                    MOVE    # 1,MEM         ;
3760          000014            MICPC=MICPC+1
3761  015760  002401            .WORD   .$.
3762  015762                    MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
3763          000015            MICPC=MICPC+1
3764  015762  041227            .WORD   .$.
3765  015764                    CALL    ERROR                   ;REPORT DATA ERROR.
3766  015764                    MOVE    # <MICPC+3>,BREG
3767          000016            MICPC=MICPC+1
3768  015764  000420            .WORD   .$.
3769  015766                    SBR     ERROR
3770          000017            MICPC=MICPC+1
3771  015766  104400            .WORD   .$.
3772  015770                    MOVE    SPAD <4>,BREG           ;RESTORE BREG...
3773          000020            MICPC=MICPC+1
3774  015770  060604            .WORD   .$.
3775  015772                    SBR     2$                      ;LOOP ON ERROR...
3776          000021            MICPC=MICPC+1
3777  015772  100403            .WORD   .$.
3778  015774            3$:     CALL    SCP1                    ;IS LOOP DATA SET.
3779  015774                    MOVE    # <MICPC+3>,BREG
3780          000022            MICPC=MICPC+1
3781  015774  000424            .WORD   .$.
3782  015776                    SBR     SCP1
3783          000023            MICPC=MICPC+1
3784  015776  104427            .WORD   .$.
3785  016000                    MOVE    SPAD <4>,BREG
3786          000024            MICPC=MICPC+1
3787  016000  060604            .WORD   .$.
3788  016002                    SBR     2$                      ;YES, DO IT.
3789          000025            MICPC=MICPC+1
3790  016002  100403            .WORD   .$.
3791  016004                    MOVE    SPAD <4>,BREG
3792          000026            MICPC=MICPC+1
3793  016004  060604            .WORD   .$.
3794  016006                    SHFRT                           ;NO, CONTINUE...
3795          000027            MICPC=MICPC+1
3796  016006  061620            .WORD   .SBR!.SELB!.DBRSH
3797  016010                    BR7     4$                      ;IS IT DONE?...
3798          000030            MICPC=MICPC+1
3799  016010  103432            .WORD   .$.
3800  016012                    SBR     2$                      ;NO, CONTINUE...
3801          000031            MICPC=MICPC+1
3802  016012  100403            .WORD   .$.
3803  016014            4$:
3804  016014            SFLOT   OUT0,INP0,1,0,0,11$,12$,13$,14$
3805  016014            11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
3806          000032            MICPC=MICPC+1
3807  016014  000577            .WORD   .$.
3808  016016            12$:
3809  016016                    MOVE    BREG,OUT0 <1>   ;SET THE BIT.
3810          000033            MICPC=MICPC+1
3811  016016  062221            .WORD   .$.
```

```
3812  016020                    MOVE    INPO <1>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
3813        000034              MICPC=MICPC+1
3814  016020  023020            .WORD   .$.
3815  016022                    MOVE    BREG,SPAD <4>
3816        000035              MICPC=MICPC+1
3817  016022  063224            .WORD   .$.
3818  016024                    $IFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...
3819
3820
3821  016024                    SUB2C   SPAD <0>,BREG,NOP
3822        000036              MICPC=MICPC+1
3823  016024  060360            .WORD   .$.
3824  016026                    BZ      13$
3825        000037              MICPC=MICPC+1
3826  016026  101452            .WORD   .$.
3827  016030                    MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
3828        000040              MICPC=MICPC+1
3829  016030  061224            .WORD   .$.
3830  016032                    MOVE    INPO <1>,OUT1 <CSR5>;BAD DATA...
3831        000041              MICPC=MICPC+1
3832  016032  021025            .WORD   .$.
3833  016034                    MOVE    # 1,MEM                 ;TYPE OF ERROR...
3834        000042              MICPC=MICPC+1
3835  016034  002401            .WORD   .$.
3836  016036                    MOVE    MEM,OUT1 <CSR3>         ;
3837        000043              MICPC=MICPC+1
3838  016036  041223            .WORD   .$.
3839  016040                    MOVE    # 1,MEM         ;
3840        000044              MICPC=MICPC+1
3841  016040  002401            .WORD   .$.
3842  016042                    MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
3843        000045              MICPC=MICPC+1
3844  016042  041227            .WORD   .$.
3845  016044                    CALL    ERROR                   ;REPORT DATA ERROR.
3846  016044                    MOVE    # <MICPC+3>,BREG
3847        000046              MICPC=MICPC+1
3848  016044  000450            .WORD   .$.
3849  016046                    SBR     ERROR
3850        000047              MICPC=MICPC+1
3851  016046  104400            .WORD   .$.
3852  016050                    MOVE    SPAD <4>,BREG           ;RESTORE BREG...
3853        000050              MICPC=MICPC+1
3854  016050  060604            .WORD   .$.
3855  016052                    SBR     12$                     ;LOOP ON ERROR...
3856        000051              MICPC=MICPC+1
3857  016052  100433            .WORD   .$.
3858  016054              13$:  CALL    SCP1                    ;IS LOOP DATA SET.
3859  016054                    MOVE    # <MICPC+3>,BREG
3860        000052              MICPC=MICPC+1
3861  016054  000454            .WORD   .$.
3862  016056                    SBR     SCP1
3863        000053              MICPC=MICPC+1
3864  016056  104427            .WORD   .$.
3865  016060                    MOVE    SPAD <4>,BREG
3866        000054              MICPC=MICPC+1
3867  016060  060604            .WORD   .$.
```

```
016062                          SBR     12$                     ;YES, DO IT.
        000055                  MICPC=MICPC+1
016062  100433                  .WORD   .$.
016064                          MOVE    SPAD <4>,BREG
        000056                  MICPC=MICPC+1
016064  060604                  .WORD   .$.
016066                          SWFBRT                          ;NO, CONTINUE...
        000057                  MICPC=MICPC+1
016066  061620                  .WORD   .SBR!.SELB!.DBRSH
016070                          BR7     12$                     ;
        000060                  MICPC=MICPC+1
016070  103433                  .WORD   .$.
016072                  14$:
016072                          CALL    SCPE
016072                          MOVE    # <MICPC+3>,BREG
        000061                  MICPC=MICPC+1
016072  000463                  .WORD   .$.
016074                          SBR     SCPE
        000062                  MICPC=MICPC+1
016074  104454                  .WORD   .$.
016076                          SBR     21$
        000063                  MICPC=MICPC+1
016076  100400                  .WORD   .$.
016100                  SFLT1   OUTO,INPO,2,0
016100                  SXZ


                                ;************************** TEST 11 **************************
                                ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
                                ;* FLOAT A 1 THROUGH REGISTER OUTO <2>
                                ;* FLOAT A 0THROUGH REGISTER OUTO <2>
016100                  $XZ
                                ;;*********************************************************************

016100                  $TSTN
                                ;  TEST 11
                                ;-------------
016100  012737  000011  001202  TST11:  MOV     #11,$TSTNM              ; LOAD THE NO. OF THIS TEST
016106  012737  016304  001442          MOV     #TST12,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
016114  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
016120  016134                          MCT11
016122  104022                          ERROR   22                      ;TIME OUT ERROR...
016124  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
016130  000177  163306                  JMP     @NEXT                   ;GO TO NEXT TEST...
016134                  MCT11:
016134                  21$:
016134                          MOVE    # 0,BREG                ;SET TO CLEAR SPAD 16
        000000                  MICPC=MICPC+1
016134  000400                  .WORD   .$.
016136                          MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
        000001                  MICPC=MICPC+1
016136  063236                  .WORD   .$.
016140                  $FLOT   OUTO,INPO,2,0,1,1$,2$,3$,4$
016140                  1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
        000002                  MICPC=MICPC+1
```

```
3924  016140  000600                    .WORD   .$.
3925  016142                    2$:
3926  016142                            MOVE    BREG,OUT0 <2>   ;SET THE BIT.
3927          000003                    MICPC=MICPC+1
3928  016142  062222                    .WORD   .$.
3929  016144                            MOVE    INP0 <2>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
3930          000004                    MICPC=MICPC+1
3931  016144  023040                    .WORD   .$.
3932  016146                            MOVE    BREG,SPAD <4>
3933          000005                    MICPC=MICPC+1
3934  016146  063224                    .WORD   .$.
3935  016150                            $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...
3936
3937
3938  016150                            SUB2C   SPAD <0>,BREG,NOP
3939          000006                    MICPC=MICPC+1
3940  016150  060360                    .WORD   .$.
3941  016152                            BZ      3$
3942          000007                    MICPC=MICPC+1
3943  016152  101422                    .WORD   .$.
3944  016154                            MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
3945          000010                    MICPC=MICPC+1
3946  016154  061224                    .WORD   .$.
3947  016156                            MOVE    INP0 <2>,OUT1 <CSR5>;BAD DATA...
3948          000011                    MICPC=MICPC+1
3949  016156  021045                    .WORD   .$.
3950  016160                            MOVE    # 1,MEM                 ;TYPE OF ERROR...
3951          000012                    MICPC=MICPC+1
3952  016160  002401                    .WORD   .$.
3953  016162                            MOVE    MEM,OUT1 <CSR3>         ;
3954          000013                    MICPC=MICPC+1
3955  016162  041223                    .WORD   .$.
3956  016164                            MOVE    # 2,MEM         ;
3957          000014                    MICPC=MICPC+1
3958  016164  002402                    .WORD   .$.
3959  016166                            MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
3960          000015                    MICPC=MICPC+1
3961  016166  041227                    .WORD   .$.
3962  016170                            CALL    ERROR                   ;REPORT DATA ERROR.
3963  016170                            MOVE    # <MICPC+3>,BREG
3964          000016                    MICPC=MICPC+1
3965  016170  000420                    .WORD   .$.
3966  016172                            $BR     EROR
3967          000017                    MICPC=MICPC+1
3968  016172  104400                    .WORD   .$.
3969  016174                            MOVE    SPAD <4>,BREG           ;RESTORE BREG...
3970          000020                    MICPC=MICPC+1
3971  016174  060604                    .WORD   .$.
3972  016176                            $BR     2$                      ;LOOP ON ERROR...
3973          000021                    MICPC=MICPC+1
3974  016176  100403                    .WORD   .$.
3975  016200                    3$:     CALL    SCP1                    ;IS LOOP DATA SET.
3976  016200                            MOVE    # <MICPC+3>,BREG
3977          000022                    MICPC=MICPC+1
3978  016200  000424                    .WORD   .$.
3979  016202                            $BR     SCP1
```

```
                                MICPC=MICPC+1
                                .WORD   .S.
                                MOVE    SPAD <4>,BREG
                                MICPC=MICPC+1
                                .WORD   .S.
                                SBR     2S                      ;YES, DO IT.
                                MICPC=MICPC+1
                                .WORD   .S.
                                MOVE    SPAD <4>,BREG
                                MICPC=MICPC+1
                                .WORD   .S.
                                SHFBRT                          ;NO, CONTINUE...
                                MICPC=MICPC+1
                                .WORD   .SBR!.SELB!.DBRSH
                                BB7     4S                      ;IS IT DONE?...
                                MICPC=MICPC+1
                                .WORD   .S.
                                SBR     2S                      ;NO, CONTINUE...
                                MICPC=MICPC+1
                                .WORD   .S.
                        4S:
                        SFLOT   OUT0,INP0,2,0,0,11$,12$,13$,14$
                        11$:    MOVE    # 177,BREG      ;START WITH BIT 7.
                                MICPC=MICPC+1
                                .WORD   .S.
                        12$:
                                MOVE    BREG,OUT0 <2>   ;SET THE BIT.
                                MICPC=MICPC+1
                                .WORD   .S.
                                MOVE    INP0 <2>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
                                MICPC=MICPC+1
                                .WORD   .S.
                                MOVE    BREG,SPAD <4>
                                MICPC=MICPC+1
                                .WORD   .S.
                                SIFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...


                                SUB2C   SPAD <0>,BREG,NOP
                                MICPC=MICPC+1
                                .WORD   .S.
                                BZ      13$
                                MICPC=MICPC+1
                                .WORD   .S.
                                MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
                                MICPC=MICPC+1
                                .WORD   .S.
                                MOVE    INP0 <2>,OUT1 <CSR5>;BAD DATA...
                                MICPC=MICPC+1
                                .WORD   .S.
                                MOVE    # 1,MEM                 ;TYPE OF ERROR...
                                MICPC=MICPC+1
                                .WORD   .S.
                                MOVE    MEM,OUT1 <CSR3>         ;
                                MICPC=MICPC+1
                                .WORD   .S.
```

```
4036 016244                     MOVE    # 2,MEM         ;
4037                            M(CPC=MICPC+1
4038 016244 000044              .WORD   .$.
4039 016246 002402              MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
4040                            MICPC=MICPC+1
4041 016246 000045              .WORD   .$.
4042 016250 041227              CALL    ERROR                   ;REPORT DATA ERROR.
4043 016250                     MOVE    # (MICPC+3),BREG
4044                            MICPC=MICPC+1
4045 016250 000046              .WORD   .$.
4046 016252 000450              SBR     ERROR
4047                            MICPC=MICPC+1
4048 016252 000047              .WORD   .$.
4049 016254 104400              MOVE    SPAD (4),BREG           ;RESTORE BREG...
4050                            MICPC=MICPC+1
4051 016254 000050              .WORD   .$.
4052 016256 060604              SBR     135                     ;LOOP ON ERROR...
4053                            MICPC=MICPC+1
4054 016256 000051              .WORD   .$.
4055 016260 100433      135:    CALL    SCP1                    ;IS LOOP DATA SET.
4056 016260                     MOVE    # (MICPC+3),BREG
4057                            MICPC=MICPC+1
4058 016260 000052              .WORD   .$.
4059 016262 000454              SBR     SCP1
4060                            MICPC=MICPC+1
4061 016262 000053              .WORD   .$.
4062 016264 104427              MOVE    SPAD (4),BREG
4063                            MICPC=MICPC+1
4064 016264 000054              .WORD   .$.
4065 016266 060604              SBR     125                     ;YES, DO IT.
4066                            MICPC=MICPC+1
4067 016266 000055              .WORD   .$.
4068 016270 100433              MOVE    SPAD (4),BREG
4069                            MICPC=MICPC+1
4070 016270 000056              .WORD   .$.
4071 016272 060604              SHFBRT                          ;NO, CONTINUE...
4072                            MICPC=MICPC+1
4073 016272 000057              .WORD   .SBR!.SELB!.DBRSH
4074 016274 061620              BB7     125                     ;
4075                            MICPC=MICPC+1
4076 016274 000060              .WORD   .$.
4077 016276 103433      145:
4078 016276                     CALL    SCPE
4079 016276                     MOVE    # (MICPC+3),BREG
4080                            MICPC=MICPC+1
4081 016276 000061              .WORD   .$.
4082 016300 000463              SBR     SCPE
4083                            MICPC=MICPC+1
4084 016300 000062              .WORD   .$.
4085 016302 104454              SBR     215
4086                            MICPC=MICPC+1
4087 016302 000063              .WORD   .$.
4088 016304 100400      SFLT1   OUTO,INPO,3,0
4089 016304             SXZ
4090
4091
```

```
                                        ;*************************** TEST 12 ***************************
                                        ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
                                        ;* FLOAT A 1 THROUGH REGISTER OUTO <3>
                                        ;* FLOAT A 0THROUGH REGISTER OUTO <3>
016304                          $XZ
                                        ;;*************************************************************************

016304                          $TSTN
                                ;  TEST 12
                                ;-------------
016304 012737 000012 001202     TST12:  MOV     #12,$TSTNM              ; LOAD THE NO. OF THIS TEST
016312 012737 016510 001442             MOV     #TST13,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
016320 004737 035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
016324 016340                           MCT12
016326 104022                           ERROR   22                      ;TIME OUT ERROR...
016330 012706 001200                    MOV     #STACK,SP               ;RESET STACK...
016334 000177 163102                    JMP     @NEXT                   ;GO TO NEXT TEST...
016340                          MCT12:
016340                          21$:
016340                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
000000                                  MICPC=MICPC+1
016340 000400                           .WORD   .$.
016342                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
000001                                  MICPC=MICPC+1
016342 063236                           .WORD   .$.
016344                          $FLOT   OUTO,INPO,3,0,1,1$,2$,3$,4$
016344                          1$:     MOVE    # 200,BREG      ;START WITH BIT 7.
000002                                  MICPC=MICPC+1
016344 000600                           .WORD   .$.
016346                          2$:
016346                                  MOVE    BREG,OUTO <3>   ;SET THE BIT.
000003                                  MICPC=MICPC+1
016346 062223                           .WORD   .$.
016350                                  MOVE    INPO <3>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
000004                                  MICPC=MICPC+1
016350 023060                           .WORD   .$.
016352                                  MOVE    BREG,SPAD <4>
000005                                  MICPC=MICPC+1
016352 063224                           .WORD   .$.
016354                                  $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...


016354                                  SUB2C   SPAD <0>,BREG,NOP
000006                                  MICPC=MICPC+1
016354 060360                           .WORD   .$.
016356                                  BZ      3$
000007                                  MICPC=MICPC+1
016356 101422                           .WORD   .$.
016360                                  MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
000010                                  MICPC=MICPC+1
016360 061224                           .WORD   .$.
016362                                  MOVE    INPO <3>,OUT1 <CSR5>;BAD DATA...
000011                                  MICPC=MICPC+1
016362 021065                           .WORD   .$.
016364                                  MOVE    # 1,MEM                 ;TYPE OF ERROR...
```

```
4148                000012                  MICPC=MICPC+1
4149  016364       002401                  .WORD   .$.
4150  016366                               MOVE    MEM,OUT1 <CSR3>         ;
4151                000013                  MICPC=MICPC+1
4152  016366       041223                  .WORD   .$.
4153  016370                               MOVE    # 3,MEM         ;
4154                000014                  MICPC=MICPC+1
4155  016370       002403                  .WORD   .$.
4156  016372                               MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
4157                000015                  MICPC=MICPC+1
4158  016372       041227                  .WORD   .$.
4159  016374                               CALL    EROR                    ;REPORT DATA ERROR.
4160  016374                               MOVE    # <MICPC+3>,BREG
4161                000016                  MICPC=MICPC+1
4162  016374       000420                  .WORD   .$.
4163  016376                               SBR     EROR
4164                000017                  MICPC=MICPC+1
4165  016376       104400                  .WORD   .$.
4166  016400                               MOVE    SPAD <4>,BREG           ;RESTORE BREG...
4167                000020                  MICPC=MICPC+1
4168  016400       060604                  .WORD   .$.
4169  016402                               SBR     2$                      ;LOOP ON ERROR...
4170                000021                  MICPC=MICPC+1
4171  016402       100403                  .WORD   .$.
4172  016404                        3$:    CALL    SCP1                    ;IS LOOP DATA SET.
4173  016404                               MOVE    # <MICPC+3>,BREG
4174                000022                  MICPC=MICPC+1
4175  016404       000424                  .WORD   .$.
4176  016406                               SBR     SCP1
4177                000023                  MICPC=MICPC+1
4178  016406       104427                  .WORD   .$.
4179  016410                               MOVE    SPAD <4>,BREG
4180                000024                  MICPC=MICPC+1
4181  016410       060604                  .WORD   .$.
4182  016412                               SBR     2$                      ;YES, DO IT.
4183                000025                  MICPC=MICPC+1
4184  016412       100403                  .WORD   .$.
4185  016414                               MOVE    SPAD <4>,BREG
4186                000026                  MICPC=MICPC+1
4187  016414       060604                  .WORD   .$.
4188  016416                               SHFBRT                          ;NO, CONTINUE...
4189                000027                  MICPC=MICPC+1
4190  016416       061620                  .WORD   .SBR!.SELB!.DBRSH
4191  016420                               BR7     4$                      ;IS IT DONE?...
4192                000030                  MICPC=MICPC+1
4193  016420       103432                  .WORD   .$.
4194  016422                               SBR     2$                      ;NO, CONTINUE...
4195                000031                  MICPC=MICPC+1
4196  016422       100403                  .WORD   .$.
4197  016424                        4$:
4198  016424                        SFLOT   OUT0,INP0,3,0,0,11$,12$,13$,14$
4199  016424                        11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
4200                000032                  MICPC=MICPC+1
4201  016424       000577                  .WORD   .$.
4202  016426                        12$:
4203  016426                               MOVE    BREG,OUT0 <3>   ;SET THE BIT.
```

```
                                MICPC=MICPC+1
                                .WORD   .S.
016426 062223                   MOVE    INPO (3),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
016430
                                MICPC=MICPC+1
000034                          .WORD   .S.
016430 023060                   MOVE    BREG,SPAD (4)
016432
                                MICPC=MICPC+1
000035                          .WORD   .S.
016432 063224                   $IFEQ   BREG,SPAD (0)  135      ;CHECK THE DATA...
016434

016434                          SUB2C   SPAD (0),BREG,NOP
                                MICPC=MICPC+1
000036                          .WORD   .S.
016434 060360                   BR      135
016436
                                MICPC=MICPC+1
000037                          .WORD   .S.
016436 101452                   MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
016440
                                MICPC=MICPC+1
000040                          .WORD   .S.
016440 061224                   MOVE    INPO (3),OUT1 (CSR5);BAD DATA...
016442
                                MICPC=MICPC+1
000041                          .WORD   .S.
016442 021065                   MOVE    # 1,MEM                 ;TYPE OF ERROR...
016444
                                MICPC=MICPC+1
000042                          .WORD   .S.
016444 002401                   MOVE    MEM,OUT1 (CSR3)         ;
016446
                                MICPC=MICPC+1
000043                          .WORD   .S.
016446 041223                   MOVE    # 3,MEM         ;
016450
                                MICPC=MICPC+1
000044                          .WORD   .S.
016450 002403                   MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
016452
                                MICPC=MICPC+1
000045                          .WORD   .S.
016452 041227                   CALL    ERROR                   ;REPORT DATA ERROR.
016454
016454                          MOVE    # (MICPC+3),BREG
                                MICPC=MICPC+1
000046                          .WORD   .S.
016454 000450                   SBR     ERROR
016456
                                MICPC=MICPC+1
000047                          .WORD   .S.
016456 104400                   MOVE    SPAD (4),BREG           ;RESTORE BREG...
016460
                                MICPC=MICPC+1
000050                          .WORD   .S.
016460 060604                   SBR     125                     ;LOOP ON ERROR...
016462
                                MICPC=MICPC+1
000051                          .WORD   .S.
016462 100433           135:    CALL    SCP1                    ;IS LOOP DATA SET.
016464
016464                          MOVE    # (MICPC+3),BREG
                                MICPC=MICPC+1
000052                          .WORD   .S.
016464 000454                   SBR     SCP1
016466
                                MICPC=MICPC+1
000053                          .WORD   .S.
016466 104427                   MOVE    SPAD (4),BREG
016470
```

```
                    000054                  MICPC=MICPC+1
016470  060604                              .WORD   .$.
016472                                      SBR     I2$                     ;YES, DO IT.
                    000055                  MICPC=MICPC+1
016472  100433                              .WORD   .$.
016474                                      MOVE    SPAD <4>,BREG
                    000056                  MICPC=MICPC+1
016474  060604                              .WORD   .$.
016476                                      SHFBRT                          ;NO, CONTINUE...
                    000057                  MICPC=MICPC+1
016476  061620                              .WORD   .SBR!.SELB!.DBRSH
016500                                      BB7     I2$                     ;
                    000060                  MICPC=MICPC+1
016500  103433                              .WORD   .$.
016502                              14$:
016502                                      CALL    SCPE
016502                                      MOVE    @ <MICPC+3>,BREG
                    000061                  MICPC=MICPC+1
016502  000463                              .WORD   .$.
016504                                      SBR     SCPE
                    000062                  MICPC=MICPC+1
016504  104454                              .WORD   .$.
016506                                      SBR     21$
                    000063                  MICPC=MICPC+1
016506  100400                              .WORD   .$.
016510                              SFLT1   OUTO,INPO,4,0
016510                                      SXZ


                                            ;*************************** TEST 13 ****************************
                                            ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
                                            ;* FLOAT A 1 THROUGH REGISTER OUTO <4>
                                            ;* FLOAT A 0THROUGH REGISTER OUTO <4>
016510                              SXZ     ;;*************************************************************************

016510                              STSTN

                                            ;  TEST 13
                                            ;--------------
016510  012737  000013  001202      TST13:  MOV     #13,STSTNM              ; LOAD THE NO. OF THIS TEST
016516  012737  016714  001442              MOV     #TST14,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                    ;R1 CONTAINS BASE KMC11 ADDRESS
016524  004737  035536                      JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
016530  016544                              MCT13
016532  104022                              ERROR   22                      ;TIME OUT ERROR...
016534  012706  001200                      MOV     #STACK,SP               ;RESET STACK...
016540  000177  162676                      JMP     @NEXT                   ;GO TO NEXT TEST...
016544                              MCT13:
016544                              21$:
016544                                      MOVE    @ 0,BREG        ;SET TO CLEAR SPAD 16
                    000000                  MICPC=MICPC+1
016544  000400                              .WORD   .$.
016546                                      MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
                    000001                  MICPC=MICPC+1
016546  063236                              .WORD   .$.
016550                              SFLOT   OUTO,INPO,4,0,1,1$,2$,3$,4$
```

```
016550                  1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
                             MICPC=MICPC+1
000002                       .WORD   .$.
016550 000600
                        2$:
                             MOVE    BREG,OUTO <4>   ;SET THE BIT.
                             MICPC=MICPC+1
000003                       .WORD   .$.
016552 062224                MOVE    INPO <4>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
016554                       MICPC=MICPC+1
000004                       .WORD   .$.
016554 023100                MOVE    BREG,SPAD <4>
016556                       MICPC=MICPC+1
000005                       .WORD   .$.
016556 063224                SIFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...
016560
                             SUB2C   SPAD <0>,BREG,NOP
016560                       MICPC=MICPC+1
000006                       .WORD   .$.
016560 060360                BZ      3$
016562                       MICPC=MICPC+1
000007                       .WORD   .$.
016562 101422                MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
016564                       MICPC=MICPC+1
000010                       .WORD   .$.
016564 061224                MOVE    INPO <4>,OUT1 <CSR5>;BAD DATA...
016566                       MICPC=MICPC+1
000011                       .WORD   .$.
016566 021105                MOVE    # 1,MEM                 ;TYPE OF ERROR...
016570                       MICPC=MICPC+1
000012                       .WORD   .$.
016570 002401                MOVE    MEM,OUT1 <CSR3>         ;
016572                       MICPC=MICPC+1
000013                       .WORD   .$.
016572 041223                MOVE    # 4,MEM          ;
016574                       MICPC=MICPC+1
000014                       .WORD   .$.
016574 002404                MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
016576                       MICPC=MICPC+1
000015                       .WORD   .$.
016576 041227                CALL    ERROR                   ;REPORT DATA ERROR.
016600                       MOVE    # <MICPC+3>,BREG
016600                       MICPC=MICPC+1
000016                       .WORD   .$.
016600 000420                SBR     ERROR
016602                       MICPC=MICPC+1
000017                       .WORD   .$.
016602 104400                MOVE    SPAD <4>,BREG           ;RESTORE BREG...
016604                       MICPC=MICPC+1
000020                       .WORD   .$.
016604 060604                SBR     2$                      ;LOOP ON ERROR...
016606                       MICPC=MICPC+1
000021                       .WORD   .$.
016606 100403                CALL    SCP1                    ;IS LOOP DATA SET.
016610                  3$:  MOVE    # <MICPC+3>,BREG
016610                       MICPC=MICPC+1
000022                       .WORD   .$.
```

```
 4372  016610  000424                    .WORD   .$.
 4373  016612                            $BR     SCP1
 4374          000023                    MICPC=MICPC+1
 4375  016612  104427                    .WORD   .$.
 4376  016614                            MOVE    SPAD <4>,BREG
 4377          000024                    MICPC=MICPC+1
 4378  016614  060604                    .WORD   .$.
 4379  016616                            $BR     2$                      ;YES, DO IT.
 4380          000025                    MICPC=MICPC+1
 4381  016616  100403                    .WORD   .$.
 4382  016620                            MOVE    SPAD <4>,BREG
 4383          000026                    MICPC=MICPC+1
 4384  016620  060604                    .WORD   .$.
 4385  016622                            SHFBRT                          ;NO, CONTINUE...
 4386          000027                    MICPC=MICPC+1
 4387  016622  061620                    .WORD   .SBR!.SELB!.DBRSH
 4388  016624                            BB7     4$                      ;IS IT DONE?...
 4389          000030                    MICPC=MICPC+1
 4390  016624  103432                    .WORD   .$.
 4391  016626                            $BR     2$                      ;NO, CONTINUE...
 4392          000031                    MICPC=MICPC+1
 4393  016626  100403                    .WORD   .$.
 4394  016630                    4$:
 4395  016630                    SFLOT   OUT0,INP0,4,0,0,11$,12$,13$,14$
 4396  016630                    11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
 4397          000032                    MICPC=MICPC+1
 4398  016630  000577                    .WORD   .$.
 4399  016632                    12$:
 4400  016632                            MOVE    BREG,OUT0 <4>   ;SET THE BIT.
 4401          000033                    MICPC=MICPC+1
 4402  016632  062224                    .WORD   .$.
 4403  016634                            MOVE    INP0 <4>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
 4404          000034                    MICPC=MICPC+1
 4405  016634  023100                    .WORD   .$.
 4406  016636                            MOVE    BREG,SPAD <4>
 4407          000035                    MICPC=MICPC+1
 4408  016636  063224                    .WORD   .$.
 4409  016640                            SIFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...
 4410
 4411
 4412  016640                            SUB2C   SPAD <0>,BREG,NOP
 4413          000036                    MICPC=MICPC+1
 4414  016640  060360                    .WORD   .$.
 4415  016642                            BZ      13$
 4416          000037                    MICPC=MICPC+1
 4417  016642  101452                    .WORD   .$.
 4418  016644                            MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
 4419          000040                    MICPC=MICPC+1
 4420  016644  061224                    .WORD   .$.
 4421  016646                            MOVE    INP0 <4>,OUT1 <CSR5>;BAD DATA...
 4422          000041                    MICPC=MICPC+1
 4423  016646  021105                    .WORD   .$.
 4424  016650                            MOVE    # 1,MEM                 ;TYPE OF ERROR...
 4425          000042                    MICPC=MICPC+1
 4426  016650  002401                    .WORD   .$.
 4427  016652                            MOVE    MEM,OUT1 <CSR3>         ;
```

```
4428                000043          MICPC=MICPC+1
4429  016652        041223          .WORD   .$.
4430  016654                        MOVE    @ 4,MEM         ;
4431                000044          MICPC=MICPC+1
4432  016654        002404          .WORD   .$.
4433  016656                        MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
4434                000045          MICPC=MICPC+1
4435  016656        041227          .WORD   .$.
4436  016660                        CALL    ERROR           ;REPORT DATA ERROR.
4437  016660                        MOVE    @ <MICPC+3>,BREG
4438                000046          MICPC=MICPC+1
4439  016660        000450          .WORD   .$.
4440  016662                        SBR     ERROR
4441                000047          MICPC=MICPC+1
4442  016662        104400          .WORD   .$.
4443  016664                        MOVE    SPAD <4>,BREG           ;RESTORE BREG...
4444                000050          MICPC=MICPC+1
4445  016664        060604          .WORD   .$.
4446  016666                        SBR     13$             ;LOOP ON ERROR...
4447                000051          MICPC=MICPC+1
4448  016666        100433          .WORD   .$.
4449  016670                13$:    CALL    SCP1            ;IS LOOP DATA SET.
4450  016670                        MOVE    @ <MICPC+3>,BREG
4451                000052          MICPC=MICPC+1
4452  016670        000454          .WORD   .$.
4453  016672                        SBR     SCP1
4454                000053          MICPC=MICPC+1
4455  016672        104427          .WORD   .$.
4456  016674                        MOVE    SPAD <4>,BREG
4457                000054          MICPC=MICPC+1
4458  016674        060604          .WORD   .$.
4459  016676                        SBR     12$             ;YES, DO IT.
4460                000055          MICPC=MICPC+1
4461  016676        100433          .WORD   .$.
4462  016700                        MOVE    SPAD <4>,BREG
4463                000056          MICPC=MICPC+1
4464  016700        060604          .WORD   .$.
4465  016702                        SHFBRT          ;NO, CONTINUE...
4466                000057          MICPC=MICPC+1
4467  016702        061620          .WORD   .SBR!.SELB!.DBRSH
4468  016704                        BR7     12$             ;
4469                000060          MICPC=MICPC+1
4470  016704        103433          .WORD   .$.
4471  016706                14$:
4472  016706                        CALL    SCPE
4473  016706                        MOVE    @ <MICPC+3>,BREG
4474                000061          MICPC=MICPC+1
4475  016706        000463          .WORD   .$.
4476  016710                        SBR     SCPE
4477                000062          MICPC=MICPC+1
4478  016710        104454          .WORD   .$.
4479  016712                        SBR     21$
4480                000063          MICPC=MICPC+1
4481  016712        100400          .WORD   .$.
4482  016714                SFLT1   OUT0,INP0,5,0
4483  016714                SXZ
```

```
                                        ;************************** TEST 14 **************************
                                        ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
                                        ;* FLOAT A 1 THROUGH REGISTER OUTO (5)
                                        ;* FLOAT A 0THROUGH REGISTER OUTO (5)
016714                          $XZ
                                        ;;*********************************************************************
016714                          $TSTN
                                ;  TEST 14
                                ;-------------
016714 012737 000014 001202     TST14:  MOV     #14,$TSTNM              ; LOAD THE NO. OF THIS TEST
016722 012737 017120 001442             MOV     #TST15,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
016730 004737 035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
016734 016750                   MCT14
016736 104022                           ERROR   22                      ;TIME OUT ERROR...
016740 012706 001200                    MOV     #STACK,SP               ;RESET STACK...
016744 000177 162472                    JMP     @NEXT                   ;GO TO NEXT TEST...
016750                          MCT14:
016750                          21$:
016750                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
000000                                  MICPC=MICPC+1
016750 000400                           .WORD   .$.
016752                                  MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
000001                                  MICPC=MICPC+1
016752 063236                           .WORD   .$.
016754                          SFLOT   OUTO,INPO,5,0,1,1$,2$,3$,4$
016754                          1$:     MOVE    # 200,BREG      ;START WITH BIT 7.
000002                                  MICPC=MICPC+1
016754 000600                           .WORD   .$.
016756                          2$:
016756                                  MOVE    BREG,OUTO (5)   ;SET THE BIT.
000003                                  MICPC=MICPC+1
016756 062225                           .WORD   .$.
016760                                  MOVE    INPO (5),SPAD (0)       ;GET THE "FOUND" IN SCRATCH PAD.
000004                                  MICPC=MICPC+1
016760 023120                           .WORD   .$.
016762                                  MOVE    BREG,SPAD (4)
000005                                  MICPC=MICPC+1
016762 063224                           .WORD   .$.
016764                                  SIFEQ   BREG,SPAD (0)  3$       ;CHECK THE DATA...

016764                                  SUB2C   SPAD (0),BREG,NOP
000006                                  MICPC=MICPC+1
016764 060360                           .WORD   .$.
016766                                  BZ      3$
000007                                  MICPC=MICPC+1
016766 101422                           .WORD   .$.
016770                                  MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
000010                                  MICPC=MICPC+1
016770 061224                           .WORD   .$.
016772                                  MOVE    INPO (5),OUT1 (CSR5);BAD DATA...
000011                                  MICPC=MICPC+1
```

```
4540  016772  021125                .WORD   .$.
4541  016774                        MOVE    # 1,MEM                 ;TYPE OF ERROR...
4542          000012                MICPC=MICPC+1
4543  016774  002401                .WORD   .$.
4544  016776                        MOVE    MEM,OUT1 <CSR3>         ;
4545          000013                MICPC=MICPC+1
4546  016776  041223                .WORD   .$.
4547  017000                        MOVE    # 5,MEM         ;
4548          000014                MICPC=MICPC+1
4549  017000  002405                .WORD   .$.
4550  017002                        MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
4551          000015                MICPC=MICPC+1
4552  017002  041227                .WORD   .$.
4553  017004                        CALL    ERROR                   ;REPORT DATA ERROR.
4554  017004                        MOVE    # <MICPC+3>,BREG
4555          000016                MICPC=MICPC+1
4556  017004  000420                .WORD   .$.
4557  017006                        SBR     ERROR
4558          000017                MICPC=MICPC+1
4559  017006  104400                .WORD   .$.
4560  017010                        MOVE    SPAD <4>,BREG           ;RESTORE BREG...
4561          000020                MICPC=MICPC+1
4562  017010  060604                .WORD   .$.
4563  017012                        SBR     2$                      ;LOOP ON ERROR...
4564          000021                MICPC=MICPC+1
4565  017012  100403                .WORD   .$.
4566  017014                3$:     CALL    SCP1                    ;IS LOOP DATA SET.
4567  017014                        MOVE    # <MICPC+3>,BREG
4568          000022                MICPC=MICPC+1
4569  017014  000424                .WORD   .$.
4570  017016                        SBR     SCP1
4571          000023                MICPC=MICPC+1
4572  017016  104427                .WORD   .$.
4573  017020                        MOVE    SPAD <4>,BREG
4574          000024                MICPC=MICPC+1
4575  017020  060604                .WORD   .$.
4576  017022                        SBR     2$                      ;YES, DO IT.
4577          000025                MICPC=MICPC+1
4578  017022  100403                .WORD   .$.
4579  017024                        MOVE    SPAD <4>,BREG
4580          000026                MICPC=MICPC+1
4581  017024  060604                .WORD   .$.
4582  017026                        SNFBRT                          ;NO, CONTINUE...
4583          000027                MICPC=MICPC+1
4584  017026  061620                .WORD   .SBR!.SELB!.DBRSH
4585  017030                        BR?     4$                      ;IS IT DONE?...
4586          000030                MICPC=MICPC+1
4587  017030  103432                .WORD   .$.
4588  017032                        SBR     2$                      ;NO, CONTINUE...
4589          000031                MICPC=MICPC+1
4590  017032  100403                .WORD   .$.
4591  017034                4$:
4592  017034                SFLOT   OUTO,INPO,5,0,0,11$,12$,13$,14$
4593  017034                11$:    MOVE    # 177,BREG      ;START WITH BIT 7.
4594          000032                MICPC=MICPC+1
4595  017034  000577                .WORD   .$.
```

```
017036                          125:
017036                                  MOVE    BREG,OUT0 <5>   ;SET THE BIT.
                  000033                MICPC=MICPC+1
017036  062225                          .WORD   .$.
017040                                  MOVE    INP0 <5>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
                  000034                MICPC=MICPC+1
017040  023120                          .WORD   .$.
017042                                  MOVE    BREG,SPAD <4>
                  000035                MICPC=MICPC+1
017042  063224                          .WORD   .$.
017044                                  SIFEQ   BREG,SPAD <0>  135      ;CHECK THE DATA...


017044                                  SUB2C   SPAD <0>,BREG,NOP
                  000036                MICPC=MICPC+1
017044  060360                          .WORD   .$.
017046                                  BZ      135
                  000037                MICPC=MICPC+1
017046  101452                          .WORD   .$.
017050                                  MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
                  000040                MICPC=MICPC+1
017050  061224                          .WORD   .$.
017052                                  MOVE    INP0 <5>,OUT1 <CSR5>;BAD DATA...
                  000041                MICPC=MICPC+1
017052  021125                          .WORD   .$.
017054                                  MOVE    # 1,MEM                 ;TYPE OF ERROR...
                  000042                MICPC=MICPC+1
017054  002401                          .WORD   .$.
017056                                  MOVE    MEM,OUT1 <CSR3>         ;
                  000043                MICPC=MICPC+1
017056  041223                          .WORD   .$.
017060                                  MOVE    # 5,MEM         ;
                  000044                MICPC=MICPC+1
017060  002405                          .WORD   .$.
017062                                  MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
                  000045                MICPC=MICPC+1
017062  041227                          .WORD   .$.
017064                                  CALL    ERROR                   ;REPORT DATA ERROR.
017064                                  MOVE    # <MICPC+3>,BREG
                  000046                MICPC=MICPC+1
017064  000450                          .WORD   .$.
017066                                  SBR     ERROR
                  000047                MICPC=MICPC+1
017066  104400                          .WORD   .$.
017070                                  MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                  000050                MICPC=MICPC+1
017070  060604                          .WORD   .$.
017072                                  SBR     125                     ;LOOP ON ERROR...
                  000051                MICPC=MICPC+1
017072  100433                          .WORD   .$.
017074                          135:    CALL    SCP1                    ;IS LOOP DATA SET.
017074                                  MOVE    # <MICPC+3>,BREG
                  000052                MICPC=MICPC+1
017074  000454                          .WORD   .$.
017076                                  SBR     SCP1
                  000053                MICPC=MICPC+1
```

```
017076  104427                  .WORD   .$.
017100                          MOVE    SPAD (4),BREG
                000054          MICPC=MICPC+1
017100  060604                  .WORD   .$.
017102                          SBR     14$                     ;YES, DO IT.
                000055          MICPC=MICPC+1
017102  100433                  .WORD   .$.
017104                          MOVE    SPAD (4),BREG
                000056          MICPC=MICPC+1
017104  060604                  .WORD   .$.
017106                          SHFRT                           ;NO, CONTINUE...
                000057          MICPC=MICPC+1
017106  061620                  .WORD   .SBR!.SELB!.DBRSH
017110                          BR7     12$                     ;
                000060          MICPC=MICPC+1
017110  103433                  .WORD   .$.
017112                  14$:
                                CALL    SCPE
                                MOVE    # (MICPC+3),BREG
                000061          MICPC=MICPC+1
017112  000463                  .WORD   .$.
017114                          SBR     SCPE
                000062          MICPC=MICPC+1
017114  104454                  .WORD   .$.
017116                          SBR     21$
                000063          MICPC=MICPC+1
017116  100400                  .WORD   .$.
017120                  $FLT1   OUTO,INPO,6,0
017120                  $XZ


                        ;************************** TEST 15 **************************
                        ;* MICRO PROCESSOR OUTO REGISTER WRITE/READ TEST.
                        ;* FLOAT A 1 THROUGH REGISTER OUTO (6)
                        ;* FLOAT A 0THROUGH REGISTER OUTO (6)
017120                  $XZ
                        ;:************************************************************

017120                  $TSTN

                        ;  TEST 15
                        ;-------------
017120  012737  000015  001202  TST15:  MOV     #15,$TSTNM              ; LOAD THE NO. OF THIS TEST
017126  012737  017324  001442          MOV     #TST16,NEXT             ; POINT TO THE START OF NEXT TEST.

017134  004737  035536                  JSR     PC,LDVRWT       ;R1 CONTAINS BASE KMC11 ADDRESS
017140  017154                          MCT15                   ;LOAD-VERIFY-WRIT.
017142  104022                          ERROR   22              ;TIME OUT ERROR...
017144  012706  001200                  MOV     #STACK,SP       ;RESET STACK...
017150  000177  162266                  JMP     @NEXT           ;GO TO NEXT TEST...
017154                  MCT15:
017154                  21$:
                                MOVE    # 0,BREG         ;SET TO CLEAR SPAD 16
                000000          MICPC=MICPC+1
017154  000400                  .WORD   .$.
017156                          MOVE    BREG,SPAD (16)  ;FOR RETURN ADDRESS PURPOSES...
                000001          MICPC=MICPC+1
```

```
 4708  017156  063236                  .WORD   $
 4709  017160                  $SLOT   .AUTO,INPO,6,0,1,1$,2$,3$,4$
 4710  017160                  1$:  MOVE   # 200,BREG    ;START WITH BIT 7.
 4711          000002                  MICPC=MICPC+1
 4712  017160  000600                  .WORD   .$.
 4713  017162                  2$:
 4714  017162                          MOVE    BREG,OUT0 <6>    ;SET THE BIT.
 4715          000003                  MICPC=MICPC+1
 4716  017162  062226                  .WORD   $
 4717  017164                          MOVE    INPO <6>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
 4718          000004                  MICPC=MICPC+1
 4719  017164  023140                  .WORD   $
 4720  017166                          MOVE    BREG,SPAD <4>
 4721          000005                  MICPC=MICPC+1
 4722  017166  063224                  .WORD   .$.
 4723  017170                          $IFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...
 4724
 4725
 4726  017170                          SUB2C   SPAD <0>,BREG,NOP
 4727          000006                  MICPC=MICPC+1
 4728  017170  060360                  .WORD   .$.
 4729  017172                          BZ      3$
 4730          000007                  MICPC=MICPC+1
 4731  017172  101422                  .WORD   $
 4732  017174                          MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
 4733          000010                  MICPC=MICPC+1
 4734  017174  061224                  .WORD   $
 4735  017176                          MOVE    INPO <6>,OUT1 <CSR5>;BAD DATA...
 4736          000011                  MICPC=MICPC+1
 4737  017176  021145                  .WORD   .$.
 4738  017200                          MOVE    # 1,MEM                 ;TYPE OF ERROR...
 4739          000012                  MICPC=MICPC+1
 4740  017200  002401                  .WORD   $
 4741  017202                          MOVE    MEM,OUT1 <CSR3>         ;
 4742          000013                  MICPC=MICPC+1
 4743  017202  041223                  .WORD   .$.
 4744  017204                          MOVE    # 6,MEM         ;
 4745          000014                  MICPC=MICPC+1
 4746  017204  002406                  .WORD   .$.
 4747  017206                          MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
 4748          000015                  MICPC=MICPC+1
 4749  017206  041227                  .WORD   .$.
 4750  017210                          CALL    ERROR                   ;REPORT DATA ERROR.
 4751  017210                          MOVE    # <MICPC+3),BREG
 4752          000016                  MICPC=MICPC+1
 4753  017210  000420                  .WORD   .$.
 4754  017212                          SBR     ERROR
 4755          000017                  MICPC=MICPC+1
 4756  017212  104400                  .WORD   $
 4757  017214                          MOVE    SPAD <4),BREG           ;RESTORE BREG...
 4758          000020                  MICPC=MICPC+1
 4759  017214  060604                  .WORD   .$.
 4760  017216                          SBR     2$                      ;LOOP ON ERROR...
 4761          000021                  MICPC=MICPC+1
 4762  017216  100403                  .WORD   .$.
 4763  017220                  3$:     CALL    SCP1                    ;IS LOOP DATA SET.
```

```
4764  017220                        MOVE    # <MICPC+3>,BREG
4765        000022                  MICPC=MICPC+1
4766  017220 000424                 .WORD   .$.
4767  017222                        SBR     SCP1
4768        000023                  MICPC=MICPC+1
4769  017222 104427                 .WORD   .$.
4770  017224                        MOVE    SPAD <4>,BREG
4771        000024                  MICPC=MICPC+1
4772  017224 060604                 .WORD   .$.
4773  017226                        SBR     2$                      ;YES, DO IT.
4774        000025                  MICPC=MICPC+1
4775  017226 100403                 .WORD   .$.
4776  017230                        MOVE    SPAD <4>,BREG
4777        000026                  MICPC=MICPC+1
4778  017230 060604                 .WORD   .$.
4779  017232                        SHFBRT                          ;NO, CONTINUE...
4780        000027                  MICPC=MICPC+1
4781  017232 061620                 .WORD   .SBR!.SELB!.DBRSH
4782  017234                        BR7     4$                      ;IS IT DONE?...
4783        000030                  MICPC=MICPC+1
4784  017234 103432                 .WORD   .$.
4785  017236                        SBR     2$                      ;NO, CONTINUE...
4786        000031                  MICPC=MICPC+1
4787  017236 100403                 .WORD   .$.
4788  017240                  4$:
4789  017240                  SFLOT   OUT0,INP0,6,0,0,11$,12$,13$,14$
4790  017240                  11$:  MOVE  # 177,BREG  ;START WITH BIT 7.
4791        000032                  MICPC=MICPC+1
4792  017240 000577                 .WORD   .$.
4793  017242                  12$:
4794  017242                        MOVE    BREG,OUT0 <6>   ;SET THE BIT.
4795        000033                  MICPC=MICPC+1
4796  017242 062226                 .WORD   .$.
4797  017244                        MOVE    INP0 <6>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
4798        000034                  MICPC=MICPC+1
4799  017244 023140                 .WORD   .$.
4800  017246                        MOVE    BREG,SPAD <4>
4801        000035                  MICPC=MICPC+1
4802  017246 063224                 .WORD   .$.
4803  017250                        $IFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...
4804
4805
4806  017250                        SUB2C   SPAD <0>,BREG,NOP
4807        000036                  MICPC=MICPC+1
4808  017250 060360                 .WORD   .$.
4809  017252                        BZ      13$
4810        000037                  MICPC=MICPC+1
4811  017252 101452                 .WORD   .$.
4812  017254                        MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
4813        000040                  MICPC=MICPC+1
4814  017254 061224                 .WORD   .$.
4815  017256                        MOVE    INP0 <6>,OUT1 <CSR5>;BAD DATA...
4816        000041                  MICPC=MICPC+1
4817  017256 021145                 .WORD   .$.
4818  017260                        MOVE    # 1,MEM                 ;TYPE OF ERROR...
4819        000042                  MICPC=MICPC+1
```

```
4820  017260  002401                .WORD   .$.
4821  017262                        MOVE    MEM,OUT1 <CSR3>         ;
                000043                MICPC=MICPC+1
4823  017262  041223                .WORD   .$.
4824  017264                        MOVE    # 6,MEM         ;
                000044                MICPC=MICPC+1
4826  017264  002406                .WORD   .$.
4827  017266                        MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
                000045                MICPC=MICPC+1
4829  017266  041227                .WORD   .$.
4830  017270                        CALL    ERROR                   ;REPORT DATA ERROR.
4831  017270                        MOVE    # <MICPC+3>,BREG
                000046                MICPC=MICPC+1
4833  017270  000450                .WORD   .$.
4834  017272                        $BR     ERROR
                000047                MICPC=MICPC+1
4836  017272  104400                .WORD   .$.
4837  017274                        MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                000050                MICPC=MICPC+1
4839  017274  060604                .WORD   .$.
4840  017276                        $BR     1$                      ;LOOP ON ERROR...
                000051                MICPC=MICPC+1
4842  017276  100433                .WORD   .$.
4843  017300                1$:     CALL    SCP1                    ;IS LOOP DATA SET.
4844  017300                        MOVE    # <MICPC+3>,BREG
                000052                MICPC=MICPC+1
4846  017300  000454                .WORD   .$.
4847  017302                        $BR     SCP1
                000053                MICPC=MICPC+1
4849  017302  104427                .WORD   .$.
4850  017304                        MOVE    SPAD <4>,BREG
                000054                MICPC=MICPC+1
4852  017304  060604                .WORD   .$.
4853  017306                        $BR     1$                      ;YES, DO IT.
                000055                MICPC=MICPC+1
4855  017306  100433                .WORD   .$.
4856  017310                        MOVE    SPAD <4>,BREG
                000056                MICPC=MICPC+1
4858  017310  060604                .WORD   .$.
4859  017312                        SHFBRT                          ;NO, CONTINUE...
                000057                MICPC=MICPC+1
4861  017312  061620                .WORD   .SBR!.SELB!.DBRSH
4862  017314                        BB7     1$              ;
                000060                MICPC=MICPC+1
4864  017314  103433                .WORD   .$.
4865  017316                2$:
4866  017316                        CALL    SCPE
4867  017316                        MOVE    # <MICPC+3>,BREG
                000061                MICPC=MICPC+1
4869  017316  000463                .WORD   .$.
4870  017320                        $BR     SCPE
                000062                MICPC=MICPC+1
4872  017320  104454                .WORD   .$.
4873  017322                        $BR     2$
                000063                MICPC=MICPC+1
4875  017322  100400                .WORD   .$.
```

```
017324                          $FLT1   OUT0,INP0,7,0
017324                          $XZ


                                ;*************************** TEST 16 ****************************
                                ;* MICRO PROCESSOR OUT0 REGISTER WRITE/READ TEST.
                                ;* FLOAT A 1 THROUGH REGISTER OUT0 <7>
                                ;* FLOAT A 0THROUGH REGISTER OUT0 <7>
017324                          $XZ
                                ;:*************************************************************

017324                          $TSTN
                                ;  TEST 16
                                ;-------------
017324  012737  000016  001202  TST16:  MOV     #16,$TSTNM              ; LOAD THE NO. OF THIS TEST
017332  012737  017530  001442          MOV     #TST17,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
017340  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
017344  017360                          MCT16
017346  104022                          ERROR   22                      ;TIME OUT ERROR...
017350  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
017354  000177  162062                  JMP     @NEXT                   ;GO TO NEXT TEST...
017360                          MCT16:
017360                          21$:
017360                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16
000000                                  MICPC=MICPC+1
017360  000400                          .WORD   .$.
017362                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
000001                                  MICPC=MICPC+1
017362  063236                          .WORD   .$.
017364                          SFLOT   OUT0,INP0,7,0,1,1$,2$,3$,4$
017364                          1$:  MOVE  # 200,BREG   ;START WITH BIT 7.
000002                                  MICPC=MICPC+1
017364  000600                          .WORD   .$.
017366                          2$:
017366                                  MOVE    BREG,OUT0 <7>   ;SET THE BIT.
000003                                  MICPC=MICPC+1
017366  062227                          .WORD   .$.
017370                                  MOVE    INP0 <7>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
000004                                  MICPC=MICPC+1
017370  023160                          .WORD   .$.
017372                                  MOVE    BREG,SPAD <4>
000005                                  MICPC=MICPC+1
017372  063224                          .WORD   .$.
017374                                  SIFEQ   BREG,SPAD <0>  3$       ;CHECK THE DATA...


017374                                  SUB2C   SPAD <0>,BREG,NOP
000006                                  MICPC=MICPC+1
017374  060360                          .WORD   .$.
017376                                  .BZ     3$
000007                                  MICPC=MICPC+1
017376  101422                          .WORD   .$.
017400                                  MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
000010                                  MICPC=MICPC+1
017400  061224                          .WORD   .$.
```

```
017402                          MOVE    INPO (7),OUT1 (CSR5);BAD DATA...
000011                          MICPC=MICPC+1
017402 021165                   .WORD   .$.
017404                          MOVE    # 1,MEM                 ;TYPE OF ERROR...
000012                          MICPC=MICPC+1
017404 002401                   .WORD   .$.
017406                          MOVE    MEM,OUT1 (CSR3)         ;
000013                          MICPC=MICPC+1
017406 041223                   .WORD   .$,
017410                          MOVE    # 7,MEM         ;
000014                          MICPC=MICPC+1
017410 002407                   .WORD   .$.
017412                          MOVE    MEM,OUT1 (CSR7)         ;REG. ADDRESS.
000015                          MICPC=MICPC+1
017412 041227                   .WORD   .$.
017414                          CALL    EROR                    ;REPORT DATA ERROR.
017414                          MOVE    # (MICPC+3),BREG
000016                          MICPC=MICPC+1
017414 000420                   .WORD   .$.
017416                          $BR     ERROR
000017                          MICPC=MICPC+1
017416 104400                   .WORD   .$.
017420                          MOVE    SPAD (4),BREG           ;RESTORE BREG...
000020                          MICPC=MICPC+1
017420 060604                   .WORD   .$.
017422                          $BR     2$                      ;LOOP ON ERROR...
000021                          MICPC=MICPC+1
017422 100403                   .WORD   .$.
017424                  3$:     CALL    SCP1                    ;IS LOOP DATA SET.
017424                          MOVE    # (MICPC+3),BREG
000022                          MICPC=MICPC+1
017424 000424                   .WORD   .$.
017426                          $BR     SCP1
000023                          MICPC=MICPC+1
017426 104427                   .WORD   .$.
017430                          MOVE    SPAD (4),BREG
000024                          MICPC=MICPC+1
017430 060604                   .WORD   .$.
017432                          $BR     2$                      ;YES, DO IT.
000025                          MICPC=MICPC+1
017432 100403                   .WORD   .$.
017434                          MOVE    SPAD (4),BREG
000026                          MICPC=MICPC+1
017434 060604                   .WORD   .$.
017436                          SHFBRT                          ;NO, CONTINUE...
000027                          MICPC=MICPC+1
017436 061620                   .WORD   .SBR!.SELB!.DBRSH
017440                          BB7     4$                      ;IS IT DONE?...
000030                          MICPC=MICPC+1
017440 103432                   .WORD   .$.
017442                          $BR     2$                      ;NO, CONTINUE...
000031                          MICPC=MICPC+1
017442 100403                   .WORD   .$.
017444                  4$:
017444          $FLOT   OUT0,INP0,7,0,0,11$,12$,13$,14$
017444          11$:    MOVE    # 177,BREG      ;START WITH BIT 7.
```

```
                                                MICPC=MICPC+1
017444  000577                                  .WORD   .$.
017446                                  12$:
017446                                          MOVE    BREG,OUT0 <7>   ;SET THE BIT.
                000033                          MICPC=MICPC+1
017446  062227                                  .WORD   .$.
017450                                          MOVE    INPO <7>,SPAD <0>       ;GET THE "FOUND" IN SCRATCH PAD.
                000034                          MICPC=MICPC+1
017450  023160                                  .WORD   .$.
017452                                          MOVE    BREG,SPAD <4>
                000035                          MICPC=MICPC+1
017452  063224                                  .WORD   .$.
017454                                          SIFEQ   BREG,SPAD <0>  13$      ;CHECK THE DATA...

017454                                          SUB2C   SPAD <0>,BREG,NOP
                000036                          MICPC=MICPC+1
017454  060360                                  .WORD   .$.
017456                                          BZ      1$
                000037                          MICPC=MICPC+1
017456  101452                                  .WORD   .$.
017460                                          MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
                000040                          MICPC=MICPC+1
017460  061224                                  .WORD   .$.
017462                                          MOVE    INPO <7>,OUT1 <CSR5>;BAD DATA...
                000041                          MICPC=MICPC+1
017462  021165                                  .WORD   .$.
017464                                          MOVE    #1,MEM                  ;TYPE OF ERROR...
                000042                          MICPC=MICPC+1
017464  002401                                  .WORD   .$.
017466                                          MOVE    MEM,OUT1 <CSR3>         ;
                000043                          MICPC=MICPC+1
017466  041223                                  .WORD   .$.
017470                                          MOVE    #7,MEM          ;
                000044                          MICPC=MICPC+1
017470  002407                                  .WORD   .$.
017472                                          MOVE    MEM,OUT1 <CSR7>         ;REG. ADDRESS.
                000045                          MICPC=MICPC+1
017472  041227                                  .WORD   .$.
017474                                          CALL    ERROR                   ;REPORT DATA ERROR.
017474                                          MOVE    # <MICPC+3>,BREG
                000046                          MICPC=MICPC+1
017474  000450                                  .WORD   .$.
017476                                          SBR     ERROR
                000047                          MICPC=MICPC+1
017476  104400                                  .WORD   .$.
017500                                          MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                000050                          MICPC=MICPC+1
017500  060604                                  .WORD   .$.
017502                                          SBR     12$                     ;LOOP ON ERROR...
                000051                          MICPC=MICPC+1
017502  100433                                  .WORD   .$.
017504                                  13$:    CALL    SCP1                    ;IS LOOP DATA SET.
                000052                          MOVE    # <MICPC+3>,BREG
017504  000454                                  MICPC=MICPC+1
                                                .WORD   .$.
```

```
017506                          SBR     SCP1
                    000053      MICPC=MICPC+1
017506  104427                  .WORD   .$.
017510                          MOVE    SPAD (4),BREG
                    000054      MICPC=MICPC+1
017510  060604                  .WORD   .$.
017512                          SBR     12$                     ;YES, DO IT.
                    000055      MICPC=MICPC+1
017512  100433                  .WORD   .$.
017514                          MOVE    SPAD (4),BREG
                    000056      MICPC=MICPC+1
017514  060604                  .WORD   .$.
017516                          SHFRT                           ;NO, CONTINUE...
                    000057      MICPC=MICPC+1
017516  061620                  .WORD   .SBR!.SELB!.DBRSH
017520                          BBS     12$                     ;
                    000060      MICPC=MICPC+1
017520  103433                  .WORD   .$.
017522                  14$:
017522                          CALL    SCPE
017522                          MOVE    # (MICPC+3),BREG
                    000061      MICPC=MICPC+1
017522  000463                  .WORD   .$.
017524                          SBR     SCPE
                    000062      MICPC=MICPC+1
017524  104454                  .WORD   .$.
017526                          SBR     21$
                    000063      MICPC=MICPC+1
017526  100400                  .WORD   .$.
017530                  SPROC4
017530                  SXZ


                                ;############################ TEST 17 ############################
                                ;# MICRO PROCESSOR B REGISTER TEST
                                ;# FLOAT A 1 THROUGH THE BREG.
                                ;# FLOAT A 0 THROUGH THE BREG.
                                ;#
017530                  SXZ     ;;#################################################################

017530                  STSTN
                                ;  TEST 17
                                ;--------------
017530  012737  000017  001202  TST17:  MOV     #17,$TSTNM              ; LOAD THE NO. OF THIS TEST
017536  012737  017720  001442          MOV     #TST20,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                              ;R1 CONTAINS BASE KMC11 ADDRESS
017544  004737  035536                JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
017550  017564                        MCT17
017552  104022                        ERROR   22                      ;TIME OUT ERROR...
017554  012706  001200                MOV     #STACK,SP               ;RESET STACK...
017560  000177  161656                JMP     @NEXT                   ;GO TO NEXT TEST...
017564                  MCT17:
017564                  21$:
017564                          MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16.
                    000000      MICPC=MICPC+1
```

```
5100  017564  000400                .WORD   .$.
5101  017566                        MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSE...
5102          000001                MICPC=MICPC+1
5103  017566  063236                .WORD   .$.
5104  017570                SBRFLT  1,1S,2S,3S,4S
5105  017570                1S:     MOVE    # 001,MEM               ;START WITH BIT0.
5106          000002                MICPC=MICPC+1
5107  017570  002401                .WORD   .$.
5108  017572                        MOVE    MEM,SPAD <2>            ;SET THE DATA IN SPAD0.
5109          000003                MICPC=MICPC+1
5110  017572  043222                .WORD   .$.
5111  017574                2S:     MOVE    MEM,BREG                ;SET THE BIT IN BREG.
5112          000004                MICPC=MICPC+1
5113  017574  040620                .WORD   .$.
5114  017576                        MOVE    BREG,SPAD <1>           ;PUT IT IN SPAD  <1>.
5115          000005                MICPC=MICPC+1
5116  017576  063221                .WORD   .$.
5117  017600                        $IFEQ   MEM,SPAD <1>    3S      ;COMPARE DATA...
5118
5119
5120  017600                        SUB2C   SPAD <1>,MEM,NOP
5121          000006                MICPC=MICPC+1
5122  017600  040361                .WORD   .$.
5123  017602                        BZ      3S
5124          000007                MICPC=MICPC+1
5125  017602  101416                .WORD   .$.
5126  017604                        MOVE    MEM,OUT1 <CSR4>         ;REPORT DATA ERROR.
5127          000010                MICPC=MICPC+1
5128  017604  041224                .WORD   .$.
5129  017606                        MOVE    BREG,OUT1 <CSR5>        ;  "    "     "   .
5130          000011                MICPC=MICPC+1
5131  017606  061225                .WORD   .$.
5132  017610                        MOVE    # 3,OUT1 <CSR3>         ;ERROR TYPE...
5133          000012                MICPC=MICPC+1
5134  017610  001003                .WORD   .$.
5135  017612                        CALL    ERROR1                  ;DATA ERROR...
5136  017612                        MOVE    @ <MICPC+3>,BREG
5137          000013                MICPC=MICPC+1
5138  017612  000415                .WORD   .$.
5139  017614                        SBR     ERROR1
5140          000014                MICPC=MICPC+1
5141  017614  104401                .WORD   .$.
5142  017616                        SBR     2S                      ;LOOP ON ERROR...
5143          000015                MICPC=MICPC+1
5144  017616  100404                .WORD   .$.
5145  017620                3S:     CALL    SCP11                   ;LOOP THE DATA...
5146  017620                        MOVE    @ <MICPC+3>,BREG
5147          000016                MICPC=MICPC+1
5148  017620  000420                .WORD   .$.
5149  017622                        SBR     SCP11
5150          000017                MICPC=MICPC+1
5151  017622  104430                .WORD   .$.
5152  017624                        SBR     2S                      ;YES, LOOP...
5153          000020                MICPC=MICPC+1
5154  017624  100404                .WORD   .$.
5155  017626                        MOVE    MEM,BREG
```

```
                    000021                  MICPC=MICPC+1
017626  040620                              .WORD   .$.
017630                                      $ROL    SPAD (2)                ;SET THE NEXT BIT.
                    000022                  MICPC=MICPC+1
017630  063142                      .WORD   .$.!.DSP
017632                                      $ADC    SPAD (2)
                    000023                  MICPC=MICPC+1
017632  063102                      .WORD   .$.!.DSP
017634                                      MOVE    SPAD (2),MEM            ;
                    000024                  MICPC=MICPC+1
017634  062602                              .WORD   .$.
017636                                      BR7     4$                  ;DONE...
                    000025                  MICPC=MICPC+1
017636  103427                              .WORD   .$.
017640                                      $BR     2$                  ;NO, CONTINUE...
                    000026                  MICPC=MICPC+1
017640  100404                              .WORD   .$.
017642                              4$:
017642                              $BRFLT  0,11$,12$,13$,14$
017642                              11$:    MOVE    #376,MEM                ;START WITH BIT0.
                    000027                  MICPC=MICPC+1
017642  002776                              .WORD   .$.
017644                                      MOVE    MEM,SPAD (2)            ;SET THE DATA IN SPAD0.
                    000030                  MICPC=MICPC+1
017644  043222                              .WORD   .$.
017646                              12$:    MOVE    MEM,BREG                ;SET THE BIT IN BREG.
                    000031                  MICPC=MICPC+1
017646  040620                              .WORD   .$.
017650                                      MOVE    BREG,SPAD (1)           ;PUT IT IN SPAD (1).
                    000032                  MICPC=MICPC+1
017650  063221                              .WORD   .$.
017652                                      $IFEQ   MEM,SPAD (1)    13$     ;COMPARE DATA...


017652                                      SUB2C   SPAD (1),MEM,NOP
                    000033                  MICPC=MICPC+1
017652  040361                              .WORD   .$.
017654                                      BZ      13$
                    000034                  MICPC=MICPC+1
017654  101443                              .WORD   .$.
017656                                      MOVE    MEM,OUT1 (CSR4)         ;REPORT DATA ERROR.
                    000035                  MICPC=MICPC+1
017656  041224                              .WORD   .$.
017660                                      MOVE    BREG,OUT1 (CSR5)        ;  "    "     "  .
                    000036                  MICPC=MICPC+1
017660  061225                              .WORD   .$.
017662                                      MOVE    #3,OUT1 (CSR3)          ;ERROR TYPE...
                    000037                  MICPC=MICPC+1
017662  001003                              .WORD   .$.
017664                                      CALL    ERROR1                  ;DATA ERROR...
017664                                      MOVE    # (MICPC+3),BREG
                    000040                  MICPC=MICPC+1
017664  000442                              .WORD   .$.
017666                                      $BR     ERROR1
                    000041                  MICPC=MICPC+1
017666  104401                              .WORD   .$.
```

```
017670                          SBR     12$                     ;LOOP ON ERROR...
                 000042         MICPC=MICPC+1
017670 100431                   .WORD   .$.
017672                  13$:    CALL    SCP11                   ;LOOP THE DATA...
017672                          MOVE    @ <MICPC+3>,BREG
                 000043         MICPC=MICPC+1
017672 000445                   .WORD   .$.
017674                          SBR     SCP11
                 000044         MICPC=MICPC+1  .
017674 104430                   .WORD   .$.
017676                          SBR     12$                     ;YES, LOOP...
                 000045         MICPC=MICPC+1
017676 100431                   .WORD   .$.
017700                          MOVE    MEM,BREG
                 000046         MICPC=MICPC+1
017700 040620                   .WORD   .$.
017702                          SBDL    SPAD <2>                ;SET THE NEXT BIT.
                 000047         MICPC=MICPC+1
017702 063142           .WORD   .$.!.DSP
017704                          SBDC    SPAD <2>
                 000050         MICPC=MICPC+1
017704 063102           .WORD   .$.!.DSP
017706                          MOVE    SPAD <2>,MEM            ;
                 000051         MICPC=MICPC+1
017706 062602                   .WORD   .$.
017710                  88$     12$             ;CONTINUE, IF NOT DONE...
                 000052         MICPC=MICPC+1
017710 103431                   .WORD   .$.
017712                  14$:
017712                          CALL    SCPE
017712                          MOVE    @ <MICPC+3>,BREG
                 000053         MICPC=MICPC+1
017712 000455                   .WORD   .$.
017714                          SBR     SCPE
                 000054         MICPC=MICPC+1
017714 104454                   .WORD   .$.
017716                          SBR     21$
                 000055         MICPC=MICPC+1
017716 100400                   .WORD   .$.
017720                  SSPTS1  4
017720                  SXZ


                                ;*************************** TEST 20 ****************************
                                ;* SCRATCH PAD TEST FOR SP4
                                ;* FLOAT A 1 THROUGH SCRATCH PAD 4
                                ;* FLOAT A 0 THROUGH SCRATCH PAD 4
017720                  SXZ
                                ;;*************************************************************
017720                  $TSTN
                                ;  TEST 20
                                ;--------------
017720 012737 000020 001202 TST20:  MOV     #20,$TSTNM              ; LOAD THE NO. OF THIS TEST
017726 012737 020134 001442         MOV     #TST21,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
```

```
017734 004737 035536            JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
017740 017754                   MCT20
017742 104022                   ERROR   22                      ;TIME OUT ERROR...
017744 012706 001200            MOV     #STACK,SP               ;RESET STACK...
017750 000177 161466            JMP     @NEXT                   ;GO TO NEXT TEST...
017754                  MCT20:
017754                  21$:
017754                          MOVE    # 0,BREG                ;SET TO CLEAR SPAD 16...
000000                          MICPC=MICPC+1
017754 000400                   .WORD   .$.
017756                          MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS PURPOSES.
000001                          MICPC=MICPC+1
017756 063236                   .WORD   .$.
017760                  $SPPFLT 1,1$,2$,3$,4$,4
017760                  1$:     MOVE # 200,BREG ;START WITH BIT 7.
000002                          MICPC=MICPC+1
017760 000600                   .WORD   .$.
017762                  2$:     MOVE    BREG,SPAD <4>           ;LOAD THE SCRATCH PAD...
000003                          MICPC=MICPC+1
017762 063224                   .WORD   .$.
017764                          MOVE    SPAD <4>,MEM            ;GET THE "FOUND"...
000004                          MICPC=MICPC+1
017764 062604                   .WORD   .$.
017766                          MOVE    MEM,SPAD <0>            ;
000005                          MICPC=MICPC+1
017766 043220                   .WORD   .$.
017770                          MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
000006                          MICPC=MICPC+1
017770 062620                   .WORD   .$.
017772                          SIFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...


017772                          SUB2C   SPAD <0>,BREG,NOP
000007                          MICPC=MICPC+1
017772 060360                   .WORD   .$.
017774                          BZ      3$
000010                          MICPC=MICPC+1
017774 101424                   .WORD   .$.
017776                          MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
000011                          MICPC=MICPC+1
017776 061224                   .WORD   .$.
020000                          MOVE    SPAD <0>,BREG           ;
000012                          MICPC=MICPC+1
020000 060600                   .WORD   .$.
020002                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
000013                          MICPC=MICPC+1
020002 061225                   .WORD   .$.
020004                          MOVE    # 4,BREG                ;TYPE OF ERROR...
000014                          MICPC=MICPC+1
020004 000404                   .WORD   .$.
020006                          MOVE    BREG,OUT1 <CSR3>        ;
000015                          MICPC=MICPC+1
020006 061223                   .WORD   .$.
020010                          MOVE    # 4,BREG                ;LOAD ADDRESS.
000016                          MICPC=MICPC+1
020010 000404                   .WORD   .$.
```

```
5324  020012                    MOVE    BREG,OUT1 <CSR7>        ;
5325          000017            MICPC=MICPC+1
5326  020012  061227            .WORD   .$.
5327  020014                    CALL    ERROR                   ;DATA ERROR!!
5328  020014                    MOVE    # <MICPC+3>,BREG
5329          000020            MICPC=MICPC+1
5330  020014  000422            .WORD   .$.
5331  020016                    SBR     ERROR
5332          000021            MICPC=MICPC+1
5333  020016  104400            .WORD   .$.
5334  020020                    MOVE    MEM,BREG                ;RESTORE BREG...
5335          000022            MICPC=MICPC+1
5336  020020  040620            .WORD   .$.
5337  020022                    SBR     2$                      ;LOOP ON ERROR...
5338          000023            MICPC=MICPC+1
5339  020022  100403            .WORD   .$.
5340  020024            3$:     CALL    SCP1                    ;IS LOOP DATA SET???
5341  020024                    MOVE    # <MICPC+3>,BREG
5342          000024            MICPC=MICPC+1
5343  020024  000426            .WORD   .$.
5344  020026                    SBR     SCP1
5345          000025            MICPC=MICPC+1
5346  020026  104427            .WORD   .$.
5347  020030                    MOVE    MEM,BREG                ;RESTORE BREG...
5348          000026            MICPC=MICPC+1
5349  020030  040620            .WORD   .$.
5350  020032                    SBR     2$                      ;LOOP ON DATA.
5351          000027            MICPC=MICPC+1
5352  020032  100403            .WORD   .$.
5353  020034                    MOVE    MEM,BREG                ;RESTORE BREG...
5354          000030            MICPC=MICPC+1
5355  020034  040620            .WORD   .$.
5356  020036                    SHFRT                           ;SET THE NEXT BIT...
5357          000031            MICPC=MICPC+1
5358  020036  061620            .WORD   .SBR!.SELB!.DBRSH
5359  020040                    BR7     4$                      ;BRANCH IF DONE...
5360          000032            MICPC=MICPC+1
5361  020040  103434            .WORD   .$.
5362  020042                    SBR     2$                      ;CONTINUE...
5363          000033            MICPC=MICPC+1
5364  020042  100403            .WORD   .$.
5365  020044            4$:
5366  020044            $SPFLT  0,11$,12$,13$,14$,4
5367  020044            11$:    MOVE # 177,BREG ;START WITH BIT 7.
5368          000034            MICPC=MICPC+1
5369  020044  000577            .WORD   .$.
5370  020046            12$:    MOVE    BREG,SPAD <4>   ;LOAD THE SCRATCH PAD...
5371          000035            MICPC=MICPC+1
5372  020046  063224            .WORD   .$.
5373  020050                    MOVE    SPAD <4>,MEM    ;GET THE "FOUND"...
5374          000036            MICPC=MICPC+1
5375  020050  062604            .WORD   .$.
5376  020052                    MOVE    MEM,SPAD <0>            ;
5377          000037            MICPC=MICPC+1
5378  020052  043220            .WORD   .$.
5379  020054                    MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
```

```
5380                    000040
5381  020054  062620            MICPC=MICPC+1
5382  020056                    .WORD   .$
                                $IFEQ   BREG,SPAD <0>   13$      ;IF GOOD.. CONTINUE...
5383
5384
5385  020056                    SUB2C   SPAD <0>,BREG,NOP
5386                    000041  MICPC=MICPC+1
5387  020056  060360            .WORD   .$
5388  020060                    BZ      13$
5389                    000042  MICPC=MICPC+1
5390  020060  101456            .WORD   .$
5391  020062                    MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
5392                    000043  MICPC=MICPC+1
5393  020062  061224            .WORD   .$
5394  020064                    MOVE    SPAD.<0>,BREG           ;
5395                    000044  MICPC=MICPC+1
5396  020064  060600            .WORD   .$
5397  020066                    MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
5398                    000045  MICPC=MICPC+1
5399  020066  061225            .WORD   .$
5400  020070                    MOVE    #4,BREG                 ;TYPE OF ERROR...
5401                    000046  MICPC=MICPC+1
5402  020070  000404            .WORD   .$
5403  020072                    MOVE    BREG,OUT1 <CSR3>        ;
5404                    000047  MICPC=MICPC+1
5405  020072  061223            .WORD   .$
5406  020074                    MOVE    #4,BREG                 ;LOAD ADDRESS.
5407                    000050  MICPC=MICPC+1
5408  020074  000404            .WORD   .$
5409  020076                    MOVE    BREG,OUT1 <CSR7>        ;
5410                    000051  MICPC=MICPC+1
5411  020076  061227            .WORD   .$
5412  020100                    CALL    ERROR
5413  020100                    MOVE    # <MICPC+3>,BREG        ;DATA ERROR!!
5414                    000052  MICPC=MICPC+1
5415  020100  000454            .WORD   .$
5416  020102                    $BR     ERROR
5417                    000053  MICPC=MICPC+1
5418  020102  104400            .WORD   .$
5419  020104                    MOVE    MEM,BREG                ;RESTORE BREG...
5420                    000054  MICPC=MICPC+1
5421  020104  040620            .WORD   .$
5422  020106                    $BR     12$                     ;LOOP ON ERROR...
5423                    000055  MICPC=MICPC+1
5424  020106  100435            .WORD   .$
5425  020110          13$:      CALL    SCP1                    ;IS LOOP DATA SET???
5426  020110                    MOVE    # <MICPC+3>,BREG
5427                    000056  MICPC=MICPC+1
5428  020110  000460            .WORD   .$
5429  020112                    $BR     SCP1
5430                    000057  MICPC=MICPC+1
5431  020112  104427            .WORD   .$
5432  020114                    MOVE    MEM,BREG                ;RESTORE BREG...
5433                    000060  MICPC=MICPC+1
5434  020114  040620            .WORD   .$
5435  020116                    $BR     12$                     ;LOOP ON DATA.
```

```
                                                MICPC=MICPC+1
                                                .WORD   .$.
                                                MOVE    MEM,BREG                ;RESTORE BREG...
                                                MICPC=MICPC+1
                                                .WORD   .$.
                                                SHFBRT                          ;SET THE NEXT BIT...
                                                MICPC=MICPC+1
                                                .WORD   .SBR!.SELB!.DBRSH
                                                BR7     12$                     ;BRANCH IF NOT DONE...
                                                MICPC=MICPC+1
                                                .WORD   .$.
                                        14$:
                                                CALL    SCPE
                                                MOVE    @ <MICPC+3>,BREG
                                                MICPC=MICPC+1
                                                .WORD   .$.
                                                SBR     SCPE
                                                MICPC=MICPC+1
                                                .WORD   .$.
                                                SBR     21$
                                                MICPC=MICPC+1
                                                .WORD   .$.
                                        $SPTS1  5
                                        $XZ

                                                ;************************** TEST 21 **************************
                                                ;* SCRATCH PAD TEST FOR SPS
                                                ;* FLOAT A 1 THROUGH SCRATCH PAD 5
                                                ;* FLOAT A 0 THROUGH SCRATCH PAD 5
                                        $XZ
                                                ;;*********************************************************************
                                        $TSTN
                                                ;  TEST 21
                                                ;--------------
020134  012737  000021  001202        TST21:  MOV     #21,$TSTNM              ; LOAD THE NO. OF THIS TEST
020142  012737  020350  001442                MOV     #TST22,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
020150  004737  035536                        JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
020154  020170                                MCT21
020156  104022                                ERROR   22                      ;TIME OUT ERROR...
020160  012706  001200                        MOV     #STACK,SP               ;RESET STACK...
020164  000177  161252                        JMP     @NEXT                   ;GO TO NEXT TEST...
020170                                  MCT21:
020170                                  21$:
                                                MOVE    @ 0,BREG        ;SET TO CLEAR SPAD 16...
                                                MICPC=MICPC+1
020170  000400                                  .WORD   .$.
020172                                          MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
                                                MICPC=MICPC+1
020172  063236                                  .WORD   .$.
020174                                  $SPFLT  1,15,28,35,45,5
020174                                  1$:     MOVE @ 200,BREG ;START WITH BIT 7.
                                                MICPC=MICPC+1
020174  000600                                  .WORD   .$.
```

```
020176 
       000003
020176 063225
020200 
       000004
020200 062605
020202 
       000005
020202 043220
020204 
       000006
020204 062620
020206 
020206 
       000007
020206 060360
020210 
       000010
020210 101424
020212 
       000011
020212 061224
020214 
       000012
020214 060600
020216 
       000013
020216 061225
020220 
       000014
020220 000404
020222 
       000015
020222 061223
020224 
       000016
020224 000405
020226 
       000017
020226 061227
020230 
020230 
       000020
020230 000422
020232 
       000021
020232 104400
020234 
       000022
020234 040620
020236 
       000023
020236 100403
020240 

25:     MOVE    BREG,SPAD (5)   ;LOAD THE SCRATCH PAD...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD (5),MEM    ;GET THE "FOUND"...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,SPAD (0)            ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
        MICPC=MICPC+1
        .WORD   .$.
        $IFEQ   BREG,SPAD (0)   3$      ;IF GOOD.. CONTINUE...

        SUB2C   SPAD (0),BREG,NOP
        MICPC=MICPC+1
        .WORD   .$.
        BZ      3$
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 (CSR4)        ;ELSE, REPORT ERROR...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD (0),BREG           ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 (CSR5)        ;BAD DATA...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    #4,BREG                 ;TYPE OF ERROR...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 (CSR3)        ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    #5,BREG                 ;LOAD ADDRESS.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 (CSR7)        ;
        MICPC=MICPC+1
        .WORD   .$.
        CALL    ERROR                   ;DATA ERROR!!
        MOVE    #(MICPC+3),BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     ERROR
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,BREG                ;RESTORE BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SBR     25                      ;LOOP ON ERROR...
        MICPC=MICPC+1
        .WORD   .$.
3$:     CALL    S[illegible]1           ;IS LOOP DATA SET???
```

```
020240                  MOVE    # <MICPC+3),BREG
                        MICPC=MICPC+1
000024                  .WORD   .S.
020240 000426           SBR     SCP1
020242                  MICPC=MICPC+1
000025                  .WORD   .S.
020242 104427           MOVE    MEM,BREG                ;RESTORE BREG...
020244                  MICPC=MICPC+1
000026                  .WORD   .S.
020244 040620           SBR     2$                      ;LOOP ON DATA.
020246                  MICPC=MICPC+1
000027                  .WORD   .S.
020246 100403           MOVE    MEM,BREG                ;RESTORE BREG...
020250                  MICPC=MICPC+1
000030                  .WORD   .S.
020250 040620           SHFBRT                          ;SET THE NEXT BIT...
020252                  MICPC=MICPC+1
000031                  .WORD   SBR!.SELB!.DBRSH
020252 061620           BR7     4$                      ;BRANCH IF DONE...
020254                  MICPC=MICPC+1
000032                  .WORD   .S.
020254 103434           SBR     2$                      ;CONTINUE...
020256                  MICPC=MICPC+1
000033                  .WORD   .S.
020256 100403
020260                  4$:
020260                  $SPFLT  0,11$,12$,13$,14$,5
000034                  11$:    MOVE # 177,BREG ;START WITH BIT 7.
020260 000577                   MICPC=MICPC+1
020262                          .WORD   .S.
000035                  12$:    MOVE    BREG,SPAD <5>   ;LOAD THE SCRATCH PAD...
020262 063225                   MICPC=MICPC+1
020264                          .WORD   .S.
000036                          MOVE    SPAD <5>,MEM    ;GET THE "FOUND"...
020264 062605                   MICPC=MICPC+1
020266                          .WORD   .S.
000037                          MOVE    MEM,SPAD <0>            ;
020266 043220                   MICPC=MICPC+1
020270                          .WORD   .S.
000040                          MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
020270 062620                   MICPC=MICPC+1
020272                          .WORD   .S.
                                $IFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...


020272                          SUB2C   SPAD <0>,BREG,NOP
000041                          MICPC=MICPC+1
020272 060360                   .WORD   .S.
020274                          BZ      13$
000042                          MICPC=MICPC+1
020274 101456                   .WORD   .S.
020276                          MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
000043                          MICPC=MICPC+1
020276 061224                   .WORD   .S.
020300                          MOVE    SPAD <0>,BREG           ;
000044                          MICPC=MICPC+1
020300 060600                   .WORD   .S.
```

```
020302
                000045
020302  061225          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
                        MICPC=MICPC+1
                        .WORD   .S.
020304                  MOVE    # 4,BREG                ;TYPE OF ERROR...
                000046  MICPC=MICPC+1
020304  000404          .WORD   .S.
020306                  MOVE    BREG,OUT1 <CSR3>        ;
                000047  MICPC=MICPC+1
020306  061223          .WORD   .S.
020310                  MOVE    # 5,BREG                ;LOAD ADDRESS.
                000050  MICPC=MICPC+1
020310  000405          .WORD   .S.
020312                  MOVE    BREG,OUT1 <CSR7>        ;
                000051  MICPC=MICPC+1
020312  061227          .WORD   .S.
020314                  CALL    ERROR                   ;DATA ERROR!!
020314                  MOVE    # <MICPC+3>,BREG
                000052  MICPC=MICPC+1
020314  000454          .WORD   .S.
020316                  SBR     ERROR
                000053  MICPC=MICPC+1
020316  104400          .WORD   .S.
020320                  MOVE    MEM,BREG                ;RESTORE BREG...
                000054  MICPC=MICPC+1
020320  040620          .WORD   .S.
020322                  SBR     135                     ;LOOP ON ERROR...
                000055  MICPC=MICPC+1
020322  100435          .WORD   .S.
020324          135:    CALL    SCP1                    ;IS LOOP DATA SET???
020324                  MOVE    # <MICPC+3>,BREG
                000056  MICPC=MICPC+1
020324  000460          .WORD   .S.
020326                  SBR     SCP1
                000057  MICPC=MICPC+1
020326  104427          .WORD   .S.
020330                  MOVE    MEM,BREG                ;RESTORE BREG...
                000060  MICPC=MICPC+1
020330  040620          .WORD   .S.
020332                  SBR     135                     ;LOOP ON DATA.
                000061  MICPC=MICPC+1
020332  100435          .WORD   .S.
020334                  MOVE    MEM,BREG                ;RESTORE BREG...
                000062  MICPC=MICPC+1
020334  040620          .WORD   .S.
020336                  SHFRT                           ;SET THE NEXT BIT...
                000063  MICPC=MICPC+1
020336  061620          .WORD   .SBR!.SELB!.DBRSH
020340                  BR7     135             ;BRANCH IF NOT DONE...
                000064  MICPC=MICPC+1
020340  103435          .WORD   .S.
020342          145:
020342                  CALL    SCPE
020342                  MOVE    # <MICPC+3>,BREG
                000065  MICPC=MICPC+1
020342  000467          .WORD   .S.
020344                  SBR     SCPE
```

```
5660                000066                          MICPC=MICPC+1
5661   020344       104454                          .WORD   .$.
5662   020346                                       $BR     21$
5663                000067                          MICPC=MICPC+1
5664   020346       100400                          .WORD   .$.
5665   020350                              $SPTS1   6
5666   020350                              $XZ
5667
5668
5669                                                ;*************************** TEST 22 ****************************
5670                                                ;* SCRATCH PAD TEST FOR SP6
5671                                                ;* FLOAT A 1 THROUGH SCRATCH PAD 6
5672                                                ;* FLOAT A 0 THROUGH SCRATCH PAD 6
5673   020350                              $XZ
5674                                                ;;*************************************************************
5675
5676   020350                              $TSTN
5677                                                ;  TEST 22
5678                                                ;--------------
5679   020350  012737  000022  001202    TST22:    MOV     #22,$TSTNM              ; LOAD THE NO. OF THIS TEST
5680   020356  012737  020564  001442              MOV     #TST23,NEXT             ; POINT TO THE START OF NEXT TEST.
5681                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
5682   020364  004737  035536                      JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
5683   020370  020404                              MCT22
5684   020372  104022                              ERROR   22                      ;TIME OUT ERROR...
5685   020374  012706  001200                      MOV     #STACK,SP               ;RESET STACK...
5686   020400  000177  161036                      JMP     @NEXT                   ;GO TO NEXT TEST...
5687   020404                              MCT22:
5688   020404                              21$:
5689   020404                                      MOVE    # 0,BREG                ;SET TO CLEAR SPAD 16...
5690                000000                         MICPC=MICPC+1
5691   020404       000400                         .WORD   .$.
5692   020406                                      MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
5693                000001                         MICPC=MICPC+1
5694   020406       063236                         .WORD   .$.
5695   020410                              SSPFLT  1,1$,2$,3$,4$,6
5696   020410                              1$:     MOVE # 200,BREG ;START WITH BIT 7.
5697                000002                         MICPC=MICPC+1
5698   020410       000600                         .WORD   .$.
5699   020412                              2$:     MOVE    BREG,SPAD <6>   ;LOAD THE SCRATCH PAD...
5700                000003                         MICPC=MICPC+1
5701   020412       063226                         .WORD   .$.
5702   020414                                      MOVE    SPAD <6>,MEM    ;GET THE "FOUND"...
5703                000004                         MICPC=MICPC+1
5704   020414       062606                         .WORD   .$.
5705   020416                                      MOVE    MEM,SPAD <0>            ;
5706                000005                         MICPC=MICPC+1
5707   020416       043220                         .WORD   .$.
5708   020420                                      MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
5709                000006                         MICPC=MICPC+1
5710   020420       062620                         .WORD   .$.
5711   020422                                      $IFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...
5712
5713
5714   020422                                      SUB2C   SPAD <0>,BREG,NOP
5715                000007                         MICPC=MICPC+1
```

```
5716 020422 060360                      .WORD   .$.
5717 020424                             BZ      35
5718        000010                      MICPC=MICPC+1
5719 020424 101424                      .WORD   .$.
5720 020426                             MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
5721        000011                      MICPC=MICPC+1
5722 020426 061224                      .WORD   .$.
5723 020430                             MOVE    SPAD <0>,BREG           ;
5724        000012                      MICPC=MICPC+1
5725 020430 060600                      .WORD   .$.
5726 020432                             MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
5727        000013                      MICPC=MICPC+1
5728 020432 061225                      .WORD   .$.
5729 020434                             MOVE    # 4,BREG                ;TYPE OF ERROR...
5730        000014                      MICPC=MICPC+1
5731 020434 000404                      .WORD   .$.
5732 020436                             MOVE    BREG,OUT1 <CSR3>        ;
5733        000015                      MICPC=MICPC+1
5734 020436 061223                      .WORD   .$.
5735 020440                             MOVE    # 6,BREG                ;LOAD ADDRESS.
5736        000016                      MICPC=MICPC+1
5737 020440 000406                      .WORD   .$.
5738 020442                             MOVE    BREG,OUT1 <CSR7>        ;
5739        000017                      MICPC=MICPC+1
5740 020442 061227                      .WORD   .$.
5741 020444                             CALL    ERROR                   ;DATA ERROR!!
5742 020444                             MOVE    # <MICPC+3>,BREG
5743        000020                      MICPC=MICPC+1
5744 020444 000422                      .WORD   .$.
5745 020446                             SBR     ERROR
5746        000021                      MICPC=MICPC+1
5747 020446 104400                      .WORD   .$.
5748 020450                             MOVE    MEM,BREG                ;RESTORE BREG...
5749        000022                      MICPC=MICPC+1
5750 020450 040620                      .WORD   .$.
5751 020452                             SBR     25                      ;LOOP ON ERROR...
5752        000023                      MICPC=MICPC+1
5753 020452 100403                      .WORD   .$.
5754 020454                     35:     CALL    SCP1                    ;IS LOOP DATA SET???
5755 020454                             MOVE    # <MICPC+3>,BREG
5756        000024                      MICPC=MICPC+1
5757 020454 000426                      .WORD   .$.
5758 020456                             SBR     SCP1
5759        000025                      MICPC=MICPC+1
5760 020456 104427                      .WORD   .$.
5761 020460                             MOVE    MEM,BREG                ;RESTORE BREG...
5762        000026                      MICPC=MICPC+1
5763 020460 040620                      .WORD   .$.
5764 020462                             SBR     25                      ;LOOP ON DATA.
5765        000027                      MICPC=MICPC+1
5766 020462 100403                      .WORD   .$.
5767 020464                             MOVE    MEM,BREG                ;RESTORE BREG...
5768        000030                      MICPC=MICPC+1
5769 020464 040620                      .WORD   .$.
5770 020466                             SHFBRT                          ;SET THE NEXT BIT...
5771        000031                      MICPC=MICPC+1
```

```
5772  020466  061620          .WORD   .SBR!.SELB!.DBRSH
5773  020470                  BR      4$                      ;BRANCH IF DONE...
5774          000032          MICPC=MICPC+1
5775  020470  103434          .WORD   .$.
5776  020472                  SBR     2$                      ;CONTINUE...
5777          000033          MICPC=MICPC+1
5778  020472  100403          .WORD   .$.
5779  020474              4$:
5780  020474              $SPFLT  0,11$,12$,13$,14$,6
5781  020474              11$:    MOVE  # 177,BREG ;START WITH BIT 7.
5782          000034          MICPC=MICPC+1
5783  020474  000577          .WORD   .$.
5784  020476              12$:    MOVE    BREG,SPAD <6>   ;LOAD THE SCRATCH PAD...
5785          000035          MICPC=MICPC+1
5786  020476  063226          .WORD   .$.
5787  020500                  MOVE    SPAD <6>,MEM    ;GET THE "FOUND"...
5788          000036          MICPC=MICPC+1
5789  020500  062606          .WORD   .$.
5790  020502                  MOVE    MEM,SPAD <0>            ;
5791          000037          MICPC=MICPC+1
5792  020502  043220          .WORD   .$.
5793  020504                  MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
5794          000040          MICPC=MICPC+1
5795  020504  062620          .WORD   .$.
5796  020506                  $IFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...
5797
5798
5799  020506                  SUB2C   SPAD <0>,BREG,NOP
5800          000041          MICPC=MICPC+1
5801  020506  060360          .WORD   .$.
5802  020510                  BZ      13$
5803          000042          MICPC=MICPC+1
5804  020510  101456          .WORD   .$.
5805  020512                  MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
5806          000043          MICPC=MICPC+1
5807  020512  061224          .WORD   .$.
5808  020514                  MOVE    SPAD <0>,BREG           ;
5809          000044          MICPC=MICPC+1
5810  020514  060600          .WORD   .$.
5811  020516                  MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
5812          000045          MICPC=MICPC+1
5813  020516  061225          .WORD   .$.
5814  020520                  MOVE    # 4,BREG                ;TYPE OF ERROR...
5815          000046          MICPC=MICPC+1
5816  020520  000404          .WORD   .$.
5817  020522                  MOVE    BREG,OUT1 <CSR3>        ;
5818          000047          MICPC=MICPC+1
5819  020522  061223          .WORD   .$.
5820  020524                  MOVE    # 6,BREG                ;LOAD ADDRESS.
5821          000050          MICPC=MICPC+1
5822  020524  000406          .WORD   .$.
5823  020526                  MOVE    BREG,OUT1 <CSR7>        ;
5824          000051          MICPC=MICPC+1
5825  020526  061227          .WORD   .$.
5826  020530                  CALL    ERROR                   ;DATA ERROR!!
5827  020530                  MOVE    # <MICPC+3>,BREG
```

```
                        MICPC=MICPC+1
020530  000052          .WORD   .S.
020532  000454          SBR     ERROR
                        MICPC=MICPC+1
020534  000053          .WORD   .S.
020536  104400          MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
020540  000054          .WORD   .S.
020542  040620          SBR     12$                     ;LOOP ON ERROR...
                        MICPC=MICPC+1
020544  000055          .WORD   .S.
020546  100435  135:    CALL    SCP1                    ;IS LOOP DATA SET???
                        MOVE    @ (MICPC+3),BREG
                        MICPC=MICPC+1
020550  000056          .WORD   .S.
020552  000460          SBR     SCP1
                        MICPC=MICPC+1
020554  000057          .WORD   .S.
020556  104427          MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
020560  000060          .WORD   .S.
020562  040620          SBR     12$                     ;LOOP ON DATA.
                        MICPC=MICPC+1
020564  000061          .WORD   .S.
020566  100435          MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
020570  000062          .WORD   .S.
020572  061620          SHFRT                           ;SET THE NEXT BIT...
                        MICPC=MICPC+1
020574  000063          .WORD   .SBR!.SELB!.DBRSH
020576  103435          BR      12$                     ;BRANCH IF NOT DONE...
                        MICPC=MICPC+1
020600  020600  145:    .WORD   .S.
020602  020602          CALL    SCPE
020604  020604          MOVE    @ (MICPC+3),BREG
                        MICPC=MICPC+1
020606  000065          .WORD   .S.
020610  000467          SBR     SCPE
                        MICPC=MICPC+1
020612  000066          .WORD   .S.
020614  104454          SBR     21$
                        MICPC=MICPC+1
020616  000067          .WORD   .S.
020620  100400  $SPTS1  7
020622  020622  $XZ


                        ;*************************** TEST 23 ****************************
                        ;* SCRATCH PAD TEST FOR SP7
                        ;* FLOAT A 1 THROUGH SCRATCH PAD 7
                        ;* FLOAT A 0 THROUGH SCRATCH PAD 7
020624  020624  $XZ
                        ;;************************************************************

020626  020626  $TSTN
```

```
5884                                        ;  TEST 23
5885                                        ;--------------
5886  020564  012737  000023  001202  TST23:  MOV     #23,$TSTNM              ; LOAD THE NO. OF THIS TEST
5887  020572  012737  021000  001442           MOV     #TST24,NEXT             ; POINT TO THE START OF NEXT TEST.
5888                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
5889  020600  004737  035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
5890  020604  020620                           MCT23
5891  020606  104022                           ERROR   22                      ;TIME OUT ERROR...
5892  020610  012706  001200                    MOV     #STACK,SP               ;RESET STACK...
5893  020614  000177  160622                    JMP     @NEXT                   ;GO TO NEXT TEST...
5894  020620                           MCT23:
5895  020620                           21$:
5896  020620                                    MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16...
5897  020620  000000                            MICPC=MICPC+1
5898  020622  000400                            .WORD   .$.
5899                                            MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
5900  020622  000001                            MICPC=MICPC+1
5901  020624  063236                            .WORD   .$.
5902  020626                           $SPFLT   1,1$,2$,3$,4$,7
5903  020626                           1$:      MOVE # 200,BREG ;START WITH BIT 7.
5904  020626  000002                            MICPC=MICPC+1
5905  020630  000600                            .WORD   .$.
5906  020632                           2$:      MOVE    BREG,SPAD <7>   ;LOAD THE SCRATCH PAD...
5907  020632  000003                            MICPC=MICPC+1
5908  020634  063227                            .WORD   .$.
5909  020636                                    MOVE    SPAD <7>,MEM    ;GET THE "FOUND"...
5910  020636  000004                            MICPC=MICPC+1
5911  020640  062607                            .WORD   .$.
5912  020642                                    MOVE    MEM,SPAD <0>            ;
5913  020642  000005                            MICPC=MICPC+1
5914  020644  043220                            .WORD   .$.
5915  020646                                    MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
5916  020646  000006                            MICPC=MICPC+1
5917  020650  062620                            .WORD   .$.
5918  020652                                    $IFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...
5919
5920
5921  020652                                    SUB2C   SPAD <0>,BREG,NOP
5922  020652  000007                            MICPC=MICPC+1
5923  020654  060360                            .WORD   .$.
5924  020656                                    BZ      3$
5925  020656  000010                            MICPC=MICPC+1
5926  020660  101424                            .WORD   .$.
5927  020662                                    MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
5928  020662  000011                            MICPC=MICPC+1
5929  020664  061224                            .WORD   .$.
5930  020666                                    MOVE    SPAD <0>,BREG           ;
5931  020666  000012                            MICPC=MICPC+1
5932  020670  060600                            .WORD   .$.
5933  020672                                    MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
5934  020672  000013                            MICPC=MICPC+1
5935  020674  061225                            .WORD   .$.
5936  020676                                    MOVE    # 4,BREG                ;TYPE OF ERROR...
5937  020676  000014                            MICPC=MICPC+1
5938  020700  000404                            .WORD   .$.
5939  020702                                    MOVE    BREG,OUT1 <CSR3>        ;
```

```
5940                    000015          MICPC=MICPC+1
5941 020652 061223      .WORD   .$.
5942 020654             MOVE    # 7,BREG                ;LOAD ADDRESS.
5943                    000016          MICPC=MICPC+1
5944 020654 000407      .WORD   .$.
5945 020656             MOVE    BREG,OUT1 <CSR7>        ;
5946                    000017          MICPC=MICPC+1
5947 020656 061227      .WORD   .$.
5948 020660             CALL    ERROR                   ;DATA ERROR!!
5949 020660             MOVE    # <MICPC+3>,BREG
5950                    000020          MICPC=MICPC+1
5951 020660 000422      .WORD   .$.
5952 020662             SBR     ERROR
5953                    000021          MICPC=MICPC+1
5954 020662 104400      .WORD   .$.
5955 020664             MOVE    MEM,BREG                ;RESTORE BREG...
5956                    000022          MICPC=MICPC+1
5957 020664 040620      .WORD   .$.
5958 020666             SBR     2$                      ;LOOP ON ERROR...
5959                    000023          MICPC=MICPC+1
5960 020666 100403      .WORD   .$.
5961 020670         3$: CALL    SCP1                    ;IS LOOP DATA SET???
5962 020670             MOVE    # <MICPC+3>,BREG
5963                    000024          MICPC=MICPC+1
5964 020670 000426      .WORD   .$.
5965 020672             SBR     SCP1
5966                    000025          MICPC=MICPC+1
5967 020672 104427      .WORD   .$.
5968 020674             MOVE    MEM,BREG                ;RESTORE BREG...
5969                    000026          MICPC=MICPC+1
5970 020674 040620      .WORD   .$.
5971 020676             SBR     2$                      ;LOOP ON DATA.
5972                    000027          MICPC=MICPC+1
5973 020676 100403      .WORD   .$.
5974 020700             MOVE    MEM,BREG                ;RESTORE BREG...
5975                    000030          MICPC=MICPC+1
5976 020700 040620      .WORD   .$.
5977 020702             SHFBRT                          ;SET THE NEXT BIT...
5978                    000031          MICPC=MICPC+1
5979 020702 061620      .WORD   .SBR!.SELB!.DBRSH
5980 020704             BR7     4$                      ;BRANCH IF DONE...
5981                    000032          MICPC=MICPC+1
5982 020704 103434      .WORD   .$.
5983 020706             SBR     2$                      ;CONTINUE...
5984                    000033          MICPC=MICPC+1
5985 020706 100403      .WORD   .$.
5986 020710         4$:
5987 020710         $SPFLT  0,11$,12$,13$,14$,7
5988 020710         11$:    MOVE # 177,BREG ;START WITH BIT 7.
5989                    000034          MICPC=MICPC+1
5990 020710 000577      .WORD   .$.
5991 020712         12$:    MOVE    BREG,SPAD <7>   ;LOAD THE SCRATCH PAD...
5992                    000035          MICPC=MICPC+1
5993 020712 063227      .WORD   .$.
5994 020714             MOVE    SPAD <7>,MEM    ;GET THE "FOUND"...
5995                    000036          MICPC=MICPC+1
```

```
 5996  020714  062607          .WORD   .$.
 5997  020716                  MOVE    MEM,SPAD <0>            ;
 5998          000037          MICPC=MICPC+1
 5999  020716  043220          .WORD   .$.
 6000  020720                  MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
 6001          000040          MICPC=MICPC+1
 6002  020720  062620          .WORD   .$.
 6003  020722                  $IFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...
 6004
 6005
 6006  020722                  SUB2C   SPAD <0>,BREG,NOP
 6007          000041          MICPC=MICPC+1
 6008  020722  060360          .WORD   .$.
 6009  020724                  BZ      .13$
 6010          000042          MICPC=MICPC+1
 6011  020724  101456          .WORD   .$.
 6012  020726                  MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
 6013          000043          MICPC=MICPC+1
 6014  020726  061224          .WORD   .$.
 6015  020730                  MOVE    SPAD <0>,BREG           ;
 6016          000044          MICPC=MICPC+1
 6017  020730  060600          .WORD   .$.
 6018  020732                  MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
 6019          000045          MICPC=MICPC+1
 6020  020732  061225          .WORD   .$.
 6021  020734                  MOVE    #4,BREG                 ;TYPE OF ERROR...
 6022          000046          MICPC=MICPC+1
 6023  020734  000404          .WORD   .$.
 6024  020736                  MOVE    BREG,OUT1 <CSR3>        ;
 6025          000047          MICPC=MICPC+1
 6026  020736  061223          .WORD   .$.
 6027  020740                  MOVE    #7,BREG                 ;LOAD ADDRESS.
 6028          000050          MICPC=MICPC+1
 6029  020740  000407          .WORD   .$.
 6030  020742                  MOVE    BREG,OUT1 <CSR7>        ;
 6031          000051          MICPC=MICPC+1
 6032  020742  061227          .WORD   .$.
 6033  020744                  CALL    ERROR                   ;DATA ERROR!!
 6034  020744                  MOVE    #<MICPC+3>,BREG
 6035          000052          MICPC=MICPC+1
 6036  020744  000454          .WORD   .$.
 6037  020746                  SBR     ERROR
 6038          000053          MICPC=MICPC+1
 6039  020746  104400          .WORD   .$.
 6040  020750                  MOVE    MEM,BREG                ;RESTORE BREG...
 6041          000054          MICPC=MICPC+1
 6042  020750  040620          .WORD   .$.
 6043  020752                  SBR     .12$                    ;LOOP ON ERROR...
 6044          000055          MICPC=MICPC+1
 6045  020752  100435          .WORD   .$.
 6046  020754          13$:    CALL    SCP1                    ;IS LOOP DATA SET???
 6047  020754                  MOVE    #<MICPC+3>,BREG
 6048          000056          MICPC=MICPC+1
 6049  020754  000460          .WORD   .$.
 6050  020756                  SBR     SCP1
 6051          000057          MICPC=MICPC+1
```

```
6052  020756  104427                    .WORD   .$.
6053  020760                            MOVE    MEM,BREG                ;RESTORE BREG...
6054          000060                    MICPC=MICPC+1
6055  020760  040620                    .WORD   .$.
6056  020762                            SBR     12$                     ;LOOP ON DATA.
6057          000061                    MICPC=MICPC+1
6058  020762  100435                    .WORD   .$.
6059  020764                            MOVE    MEM,BREG                ;RESTORE BREG...
6060          000062                    MICPC=MICPC+1
6061  020764  040620                    .WORD   .$.
6062  020766                            SHFBRT                          ;SET THE NEXT BIT...
6063          000063                    MICPC=MICPC+1
6064  020766  061620                    .WORD   .SBR!.SELB!.DBRSH
6065  020770                            BBS     12$                     ;BRANCH IF NOT DONE...
6066          000064                    MICPC=MICPC+1
6067  020770  103435                    .WORD   .$.
6068  020772                    14$:
6069  020772                            CALL    SCPE
6070  020772                            MOVE    # <MICPC+3>,BREG
6071          000065                    MICPC=MICPC+1
6072  020772  000467                    .WORD   .$.
6073  020774                            SBR     SCPE
6074          000066                    MICPC=MICPC+1
6075  020774  104454                    .WORD   .$.
6076  020776                            SBR     21$
6077          000067                    MICPC=MICPC+1
6078  020776  100400                    .WORD   .$.
6079  021000                    $SPTS1  10
6080  021000                    $XZ
6081
6082
6083                                    ;*************************** TEST 24 ****************************
6084                                    ;* SCRATCH PAD TEST FOR SP10
6085                                    ;* FLOAT A 1 THROUGH SCRATCH PAD 10
6086                                    ;* FLOAT A 0 THROUGH SCRATCH PAD 10
6087  021000                    $XZ
6088                                    ;:*************************************************************
6089
6090  021000                    $TSTN
6091                                    ;  TEST 24
6092                                    ;-------------
6093  021000  012737  000024  001202    TST24:  MOV     #24,$TSTNM              ; LOAD THE NO. OF THIS TEST
6094  021006  012737  021214  001442            MOV     #TST25,NEXT             ; POINT TO THE START OF NEXT TEST.
6095                                                                    ;R1 CONTAINS BASE KMC11 ADDRESS
6096  021014  004737  035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
6097  021020  021034                            MCT24
6098  021022  104022                            ERROR   22                      ;TIME OUT ERROR...
6099  021024  012706  001200                    MOV     #STACK,SP               ;RESET STACK...
6100  021030  000177  160406                    JMP     @NEXT                   ;GO TO NEXT TEST...
6101  021034                            MCT24:
6102  021034                            21$:
6103  021034                            MOVE    # 0,BREG              ;SET TO CLEAR SPAD 16...
6104          000000                    MICPC=MICPC+1
6105  021034  000400                    .WORD   .$.
6106  021036                            MOVE    BREG,SPAD <16>        ;FOR RETURN ADDRESS PURPOSES.
6107          000001                    MICPC=MICPC+1
```

```
6108  021036  063236          .WORD   .$.
6109  021040                  $SPFLT  1,1$,2$,3$,4$,10
6110  021040                  1$:     MOVE    # 200,BREG ;START WITH BIT 7.
6111          000002                  MICPC=MICPC+1
6112  021040  000600                  .WORD   .$.
6113  021042                  2$:     MOVE    BREG,SPAD <10>  ;LOAD THE SCRATCH PAD...
6114          000003                  MICPC=MICPC+1
6115  021042  063230                  .WORD   .$.
6116  021044                          MOVE    SPAD <10>,MEM   ;GET THE "FOUND"...
6117          000004                  MICPC=MICPC+1
6118  021044  062610                  .WORD   .$.
6119  021046                          MOVE    MEM,SPAD <0>            ;
6120          000005                  MICPC=MICPC+1
6121  021046  043220                  .WORD   .$.
6122  021050                          MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
6123          000006                  MICPC=MICPC+1
6124  021050  062620                  .WORD   .$.
6125  021052                          $IFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...
6126
6127
6128  021052                          SUB2C   SPAD <0>,BREG,NOP
6129          000007                  MICPC=MICPC+1
6130  021052  060360                  .WORD   .$.
6131  021054                          BZ      3$
6132          000010                  MICPC=MICPC+1
6133  021054  101424                  .WORD   .$.
6134  021056                          MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
6135          000011                  MICPC=MICPC+1
6136  021056  061224                  .WORD   .$.
6137  021060                          MOVE    SPAD <0>,BREG           ;
6138          000012                  MICPC=MICPC+1
6139  021060  060600                  .WORD   .$.
6140  021062                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
6141          000013                  MICPC=MICPC+1
6142  021062  061225                  .WORD   .$.
6143  021064                          MOVE    # 4,BREG                ;TYPE OF ERROR...
6144          000014                  MICPC=MICPC+1
6145  021064  000404                  .WORD   .$.
6146  021066                          MOVE    BREG,OUT1 <CSR3>        ;
6147          000015                  MICPC=MICPC+1
6148  021066  061223                  .WORD   .$.
6149  021070                          MOVE    # 10,BREG               ;LOAD ADDRESS.
6150          000016                  MICPC=MICPC+1
6151  021070  000410                  .WORD   .$.
6152  021072                          MOVE    BREG,OUT1 <CSR7>        ;
6153          000017                  MICPC=MICPC+1
6154  021072  061227                  .WORD   .$.
6155  021074                          CALL    ERROR                   ;DATA ERROR!!
6156  021074                          MOVE    # <MICPC+3>,BREG
6157          000020                  MICPC=MICPC+1
6158  021074  000422                  .WORD   .$.
6159  021076                          SBR     ERROR
6160          000021                  MICPC=MICPC+1
6161  021076  104400                  .WORD   .$.
6162  021100                          MOVE    MEM,BREG                ;RESTORE BREG...
6163          000022                  MICPC=MICPC+1
```

```
 6164  021100  040620                    .WORD   .$.
 6165  021102                            $BR     2$                      ;LOOP ON ERROR...
 6166          000023                    MICPC=MICPC+1
 6167  021102  100403                    .WORD   .$.
 6168  021104                    3$:     CALL    $CP1                    ;IS LOOP DATA SET???
 6169  021104                            MOVE    @ <MICPC+3>,BREG
 6170          000024                    MICPC=MICPC+1
 6171  021104  000426                    .WORD   .$.
 6172  021106                            $BR     $CP1
 6173          000025                    MICPC=MICPC+1
 6174  021106  104427                    .WORD   .$.
 6175  021110                            MOVE    MEM,BREG                ;RESTORE BREG...
 6176          000026                    MICPC=MICPC+1
 6177  021110  040620                    .WORD   .$.
 6178  021112                            $BR     2$                      ;LOOP ON DATA.
 6179          000027                    MICPC=MICPC+1
 6180  021112  100403                    .WORD   .$.
 6181  021114                            MOVE    MEM,BREG                ;RESTORE BREG...
 6182          000030                    MICPC=MICPC+1
 6183  021114  040620                    .WORD   .$.
 6184  021116                            SHFRT                           ;SET THE NEXT BIT...
 6185          000031                    MICPC=MICPC+1
 6186  021116  061620                    .WORD   .SBR!.SELB!.DBRSH
 6187  021120                            BRZ     4$                      ;BRANCH IF DONE...
 6188          000032                    MICPC=MICPC+1
 6189  021120  103434                    .WORD   .$.
 6190  021122                            $BR     2$                      ;CONTINUE...
 6191          000033                    MICPC=MICPC+1
 6192  021122  100403                    .WORD   .$.
 6193  021124                    4$:
 6194  021124                    $SPFLT  0,11$,12$,13$,14$,10
 6195  021124                    11$:    MOVE @ 177,BREG ;START WITH BIT 7.
 6196          000034                    MICPC=MICPC+1
 6197  021124  000577                    .WORD   .$.
 6198  021126                    12$:    MOVE    BREG,SPAD <10>  ;LOAD THE SCRATCH PAD...
 6199          000035                    MICPC=MICPC+1
 6200  021126  063230                    .WORD   .$.
 6201  021130                            MOVE    SPAD <10>,MEM   ;GET THE "FOUND"...
 6202          000036                    MICPC=MICPC+1
 6203  021130  062610                    .WORD   .$.
 6204  021132                            MOVE    MEM,SPAD <0>            ;
 6205          000037                    MICPC=MICPC+1
 6206  021132  043220                    .WORD   .$.
 6207  021134                            MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
 6208          000040                    MICPC=MICPC+1
 6209  021134  062620                    .WORD   .$.
 6210  021136                            $IFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...
 6211
 6212
 6213  021136                            SUB2C   SPAD <0>,BREG,NOP
 6214          000041                    MICPC=MICPC+1
 6215  021136  060360                    .WORD   .$.
 6216  021140                            BZ      13$
 6217          000042                    MICPC=MICPC+1
 6218  021140  101456                    .WORD   .$.
 6219  021142                            MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
```

```
021142 000043 061224
021144 000044 060600
021146 000045 061225
021150 000046 000404
021152 000047 061223
021154 000050 000410
021156 000051 061227
021160 000052 000454
021162 000053 104400
021164 000054 040620
021166 000055 100435
021170 000056 000460
021172 000057 104427
021174 000060 040620
021176 000061 100435
021200 000062 040620
021202 000063 061620
021204 000064 103435
021206
```

```
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <0>,BREG           ;
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 4,BREG                ;TYPE OF ERROR...
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 <CSR3>        ;
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 10,BREG               ;LOAD ADDRESS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 <CSR7>        ;
                MICPC=MICPC+1
                .WORD   .$.
                CALL    ERROR                   ;DATA ERROR!!
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     ERROR
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     1$                      ;LOOP ON ERROR...
                MICPC=MICPC+1
                .WORD   .$.
135:            CALL    SCP1                    ;IS LOOP DATA SET???
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     1$                      ;LOOP ON DATA.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SHFBRT                          ;SET THE NEXT BIT...
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                BB7     1$              ;BRANCH IF NOT DONE...
                MICPC=MICPC+1
                .WORD   .$.
145:
```

```
021206                                  CALL    SCPE
021206                                  MOVE    @ <MICPC+3>,BREG
        000065                          MICPC=MICPC+1
021206  000467                          .WORD   .$.
021210                                  $BR     SCPE
        000066                          MICPC=MICPC+1
021210  104454                          .WORD   .$.
021212                                  $BR     21$
        000067                          MICPC=MICPC+1
021212  100400                          .WORD   .$.
021214                          $SPTS1  11
021214                          $XZ


                                        ;************************** TEST 25 **************************
                                        ;* SCRATCH PAD TEST FOR SP11
                                        ;* FLOAT A 1 THROUGH SCRATCH PAD 11
                                        ;* FLOAT A 0 THROUGH SCRATCH PAD 11
021214                          $XZ     ;:**********************************************************************

021214                          $TSTN
                                ;  TEST 25
                                ;--------------
021214  012737  000025  001202  TST25:  MOV     #25,$TSTNM              ; LOAD THE NO. OF THIS TEST
021222  012737  021430  001442          MOV     #TST26,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
021230  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
021234  021250                          MCT25
021236  104022                          ERROR   22                      ;TIME OUT ERROR...
021240  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
021244  000177  160172                  JMP     @NEXT                   ;GO TO NEXT TEST...
021250                          MCT25:
021250                          21$:
021250                                  MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16...
        000000                          MICPC=MICPC+1
021250  000400                          .WORD   .$.
021252                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
        000001                          MICPC=MICPC+1
021252  063236                          .WORD   .$.
021254                          $SPFLT  1,1$,2$,3$,4$,11
021254                          1$:     MOVE # 200,BREG ;START WITH BIT 7.
        000002                          MICPC=MICPC+1
021254  000600                          .WORD   .$.
021256                          2$:     MOVE    BREG,SPAD <11>  ;LOAD THE SCRATCH PAD...
        000003                          MICPC=MICPC+1
021256  063231                          .WORD   .$.
021260                                  MOVE    SPAD <11>,MEM   ;GET THE "FOUND"...
        000004                          MICPC=MICPC+1
021260  062611                          .WORD   .$.
021262                                  MOVE    MEM,SPAD <0>            ;
        000005                          MICPC=MICPC+1
021262  043220                          .WORD   .$.
021264                                  MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
        000006                          MICPC=MICPC+1
021264  062620                          .WORD   .$.
```

```
021266                          SIFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...

021266                          SUB2C   SPAD <0>,BREG,NOP
                                MICPC=MICPC+1
021266  060360                  .WORD   .$.
021270                          BZ      3$
                                MICPC=MICPC+1
021270  101424                  .WORD   .$.
021272                          MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
                                MICPC=MICPC+1
021272  061224                  .WORD   .$.
021274                          MOVE    SPAD <0>,BREG           ;
                                MICPC=MICPC+1
021274  060600                  .WORD   .$.
021276                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
                                MICPC=MICPC+1
021276  061225                  .WORD   .$.
021300                          MOVE    #4,BREG                 ;TYPE OF ERROR...
                                MICPC=MICPC+1
021300  000404                  .WORD   .$.
021302                          MOVE    BREG,OUT1 <CSR3>        ;
                                MICPC=MICPC+1
021302  061223                  .WORD   .$.
021304                          MOVE    #11,BREG                ;LOAD ADDRESS.
                                MICPC=MICPC+1
021304  000411                  .WORD   .$.
021306                          MOVE    BREG,OUT1 <CSR7>        ;
                                MICPC=MICPC+1
021306  061227                  .WORD   .$.
021310                          CALL    ERROR                   ;DATA ERROR!!
021310                          MOVE    #<MICPC+3>,BREG
                                MICPC=MICPC+1
021310  000422                  .WORD   .$.
021312                          SBR     ERROR
                                MICPC=MICPC+1
021312  104400                  .WORD   .$.
021314                          MOVE    MEM,BREG                ;RESTORE BREG...
                                MICPC=MICPC+1
021314  040620                  .WORD   .$.
021316                          SBR     2$                      ;LOOP ON ERROR...
                                MICPC=MICPC+1
021316  100403                  .WORD   .$.
021320                  3$:     CALL    SCP1                    ;IS LOOP DATA SET???
021320                          MOVE    #<MICPC+3>,BREG
                                MICPC=MICPC+1
021320  000426                  .WORD   .$.
021322                          SBR     SCP1
                                MICPC=MICPC+1
021322  104427                  .WORD   .$.
021324                          MOVE    MEM,BREG                ;RESTORE BREG...
                                MICPC=MICPC+1
021324  040620                  .WORD   .$.
021326                          SBR     2$                      ;LOOP ON DATA.
                                MICPC=MICPC+1
021326  100403                  .WORD   .$.
```

```
021330
                000030
021330  040620
021332
                000031
021332  061620
021334
                000032
021334  103434
021336
                000033
021336  100403
021340
021340
021340
                000034
021340  000577
021342
                000035
021342  063231
021344
                000036
021344  062611
021346
                000037
021346  043220
021350
                000040
021350  062620
021352

021352
                000041
021352  060360
021354
                000042
021354  101456
021356
                000043
021356  061224
021360
                000044
021360  060600
021362
                000045
021362  061225
021364
                000046
021364  000404
021366
                000047
021366  061223
021370
                000050
021370  000411
```

```
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .S.
                SHFBRT                          ;SET THE NEXT BIT...
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                BBZ     4$                      ;BRANCH IF DONE...
                MICPC=MICPC+1
                .WORD   .S.
                SBR     2$                      ;CONTINUE...
                MICPC=MICPC+1
                .WORD   .S.
4$:
SSPFLT  0,11$,12$,13$,14$,11
11$:    MOVE  # 177,BREG ;START WITH BIT 7.
                MICPC=MICPC+1
                .WORD   .S.
12$:    MOVE    BREG,SPAD <11>  ;LOAD THE SCRATCH PAD...
                MICPC=MICPC+1
                .WORD   .S.
                MOVE    SPAD <11>,MEM   ;GET THE "FOUND"...
                MICPC=MICPC+1
                .WORD   .S.
                MOVE    MEM,SPAD <0>            ;
                MICPC=MICPC+1
                .WORD   .S.
                MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
                MICPC=MICPC+1
                .WORD   .S.
                $IFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...

                SUB2C   SPAD <0>,BREG,NOP
                MICPC=MICPC+1
                .WORD   .S.
                BZ      13$
                MICPC=MICPC+1
                .WORD   .S.
                MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
                MICPC=MICPC+1
                .WORD   .S.
                MOVE    SPAD <0>,BREG           ;
                MICPC=MICPC+1
                .WORD   .S.
                MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
                MICPC=MICPC+1
                .WORD   .S.
                MOVE    # 4,BREG                ;TYPE OF ERROR...
                MICPC=MICPC+1
                .WORD   .S.
                MOVE    BREG,OUT1 <CSR3>        ;
                MICPC=MICPC+1
                .WORD   .S.
                MOVE    # 11,BREG               ;LOAD ADDRESS.
                MICPC=MICPC+1
                .WORD   .S.
```

```
021372
                000051
021372          061227
021374
021374
                000052
021374          000454
021376
                000053
021376          104400
021400
                000054
021400          040620
021402
                000055
021402          100435
021404
021404
                000056
021404          000460
021406
                000057
021406          104427
021410
                000060
021410          040620
021412
                000061
021412          100435
021414
                000062
021414          040620
021416
                000063
021416          061620
021420
                000064
021420          103435
021422
021422
021422
                000065
021422          000467
021424
                000066
021424          104454
021426
                000067
021426          100400
021430
021430
```

```
                MOVE    BREG,OUT1 (CSR7)        ;
                MICPC=MICPC+1
                .WORD   .$.
                CALL    ERROR                   ;DATA ERROR!!
                MOVE    @ (MICPC+3),BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     ERROR
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     1$                      ;LOOP ON ERROR...
                MICPC=MICPC+1
                .WORD   .$.
13$:            CALL    SCP1                    ;IS LOOP DATA SET???
                MOVE    @ (MICPC+3),BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     1$                      ;LOOP ON DATA.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SHFBRT                          ;SET THE NEXT BIT...
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                BB7     1$                      ;BRANCH IF NOT DONE...
                MICPC=MICPC+1
                .WORD   .$.
14$:
                CALL    SCPE
                MOVE    @ (MICPC+3),BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCPE
                MICPC=MICPC+1
                .WORD   .$.
                SBR     21$
                MICPC=MICPC+1
                .WORD   .$.
$SPTS1  12
$XZ


                ;*************************** TEST 26 ****************************
                ;* SCRATCH PAD TEST FOR SP12
                ;* FLOAT A 1 THROUGH SCRATCH PAD 12
```

```
                                        ;* FLOAT A 0 THROUGH SCRATCH PAD 12
021430                          $XZ
                                        ;;*****************************************************************
021430                          $TSTN
                                        ;  TEST 26
                                        ;--------------
021430  012737  000026  001202  TST26:  MOV     #26,$TSTNM              ; LOAD THE NO. OF THIS TEST
021436  012737  021644  001442          MOV     #TST27,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
021444  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
021450  021464                          MCT26
021452  104022                          ERROR   22                      ;TIME OUT ERROR...
021454  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
021460  000177  157756                  JMP     @NEXT                   ;GO TO NEXT TEST...
021464                          MCT26:
021464                          21$:
                                        MOVE    # 0,BREG         ;SET TO CLEAR SPAD 16...
        000000                  
021464  000400                  MICPC=MICPC+1
                                .WORD   .$.
021466                          MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
        000001                  
021466  063236                  MICPC=MICPC+1
021470                          .WORD   .$.
021470                          $SPFLT  1,1$,2$,3$,4$,12
021470                          1$:     MOVE # 200,BREG ;START WITH BIT 7.
        000002                  
021470  000600                  MICPC=MICPC+1
021472                          .WORD   .$.
                                2$:     MOVE    BREG,SPAD <12>  ;LOAD THE SCRATCH PAD...
        000003                  
021472  063232                  MICPC=MICPC+1
021474                          .WORD   .$.
                                MOVE    SPAD <12>,MEM   ;GET THE "FOUND"...
        000004                  
021474  062612                  MICPC=MICPC+1
021476                          .WORD   .$.
                                MOVE    MEM,SPAD <0>            ;
        000005                  
021476  043220                  MICPC=MICPC+1
021500                          .WORD   .$.
                                MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
        000006                  
021500  062620                  MICPC=MICPC+1
021502                          .WORD   .$.
                                $IFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...

021502                          
                                SUB2C   SPAD <0>,BREG,NOP
        000007                  
021502  060360                  MICPC=MICPC+1
021504                          .WORD   .$.
                                BZ      3$
        000010                  
021504  101424                  MICPC=MICPC+1
021506                          .WORD   .$.
                                MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
        000011                  
021506  061224                  MICPC=MICPC+1
021510                          .WORD   .$.
                                MOVE    SPAD <0>,BREG           ;
        000012                  
021510  060600                  MICPC=MICPC+1
021512                          .WORD   .$.
                                MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
        000013                  MICPC=MICPC+1
```

```
021512 061225                  .WORD   .S.
021514                         MOVE    # 4,BREG                ;TYPE OF ERROR...
       000014                  MICPC=MICPC+1
021514 000404                  .WORD   .S.
021516                         MOVE    BREG,OUT1 <CSR3>        ;
       000015                  MICPC=MICPC+1
021516 061223                  .WORD   .S.
021520                         MOVE    # 12,BREG               ;LOAD ADDRESS.
       000016                  MICPC=MICPC+1
021520 000412                  .WORD   .S.
021522                         MOVE    BREG,OUT1 <CSR7>        ;
       000017                  MICPC=MICPC+1
021522 061227                  .WORD   .S.
021524                         CALL    ERROR                   ;DATA ERROR!!
021524                         MOVE    # <MICPC+3>,BREG
       000020                  MICPC=MICPC+1
021524 000422                  .WORD   .S.
021526                         SBR     ERROR
       000021                  MICPC=MICPC+1
021526 104400                  .WORD   .S.
021530                         MOVE    MEM,BREG                ;RESTORE BREG...
       000022                  MICPC=MICPC+1
021530 040620                  .WORD   .S.
021532                         SBR     2S                      ;LOOP ON ERROR...
       000023                  MICPC=MICPC+1
021532 100403                  .WORD   .S.
021534                 3S:     CALL    SCP1                    ;IS LOOP DATA SET???
021534                         MOVE    # <MICPC+3>,BREG
       000024                  MICPC=MICPC+1
021534 000426                  .WORD   .S.
021536                         SBR     SCP1
       000025                  MICPC=MICPC+1
021536 104427                  .WORD   .S.
021540                         MOVE    MEM,BREG                ;RESTORE BREG...
       000026                  MICPC=MICPC+1
021540 040620                  .WORD   .S.
021542                         SBR     2S                      ;LOOP ON DATA.
       000027                  MICPC=MICPC+1
021542 100403                  .WORD   .S.
021544                         MOVE    MEM,BREG                ;RESTORE BREG...
       000030                  MICPC=MICPC+1
021544 040620                  .WORD   .S.
021546                         SHFRT                           ;SET THE NEXT BIT...
       000031                  MICPC=MICPC+1
021546 061620                  .WORD   .SBR!.SELB!.DBRSH
021550                         BR7     4S                      ;BRANCH IF DONE...
       000032                  MICPC=MICPC+1
021550 103434                  .WORD   .S.
021552                         SBR     2S                      ;CONTINUE...
       000033                  MICPC=MICPC+1
021552 100403                  .WORD   .S.
021554                 4S:
021554                 SSPFLT  0,11S,12S,13S,14S,12
021554                 11S:    MOVE # 177,BREG ;START WITH BIT 7.
       000034                  MICPC=MICPC+1
021554 000577                  .WORD   .S.
```

```
6612  021556                  12$:    MOVE    BREG,SPAD (12)  ;LOAD THE SCRATCH PAD...
6613                                  MICPC=MICPC+1
6614  021556  000035                  .WORD   .$.
6615  021560  063232                  MOVE    SPAD (12),MEM   ;GET THE "FOUND"...
6616                                  MICPC=MICPC+1
6617  021560  000036                  .WORD   .$.
6618  021562  062612                  MOVE    MEM,SPAD (0)            ;
6619                                  MICPC=MICPC+1
6620  021562  000037                  .WORD   .$.
6621  021564  043220                  MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
6622                                  MICPC=MICPC+1
6623  021564  000040                  .WORD   .$.
6624  021566  062620                  $IFEQ   BREG,SPAD (0)   13$     ;IF GOOD.. CONTINUE...
6625
6626
6627  021566                          SUB2C   SPAD (0),BREG,NOP
6628                                  MICPC=MICPC+1
6629  021566  000041                  .WORD   .$.
6630  021570  060360                  BZ      13$
6631                                  MICPC=MICPC+1
6632  021570  000042                  .WORD   .$.
6633  021572  101456                  MOVE    BREG,OUT1 (CSR4)        ;ELSE, REPORT ERROR...
6634                                  MICPC=MICPC+1
6635  021572  000043                  .WORD   .$.
6636  021574  061224                  MOVE    SPAD (0),BREG           ;
6637                                  MICPC=MICPC+1
6638  021574  000044                  .WORD   .$.
6639  021576  060600                  MOVE    BREG,OUT1 (CSR5)        ;BAD DATA...
6640                                  MICPC=MICPC+1
6641  021576  000045                  .WORD   .$.
6642  021600  061225                  MOVE    #4,BREG                 ;TYPE OF ERROR...
6643                                  MICPC=MICPC+1
6644  021600  000046                  .WORD   .$.
6645  021602  000404                  MOVE    BREG,OUT1 (CSR3)        ;
6646                                  MICPC=MICPC+1
6647  021602  000047                  .WORD   .$.
6648  021604  061223                  MOVE    #12,BREG                ;LOAD ADDRESS.
6649                                  MICPC=MICPC+1
6650  021604  000050                  .WORD   .$.
6651  021606  000412                  MOVE    BREG,OUT1 (CSR7)        ;
6652                                  MICPC=MICPC+1
6653  021606  000051                  .WORD   .$.
6654  021610  061227                  CALL    ERROR                   ;DATA ERROR!!
6655  021610                          MOVE    #(MICPC+3),BREG
6656                                  MICPC=MICPC+1
6657  021610  000052                  .WORD   .$.
6658  021612  000454                  SBR     ERROR
6659                                  MICPC=MICPC+1
6660  021612  000053                  .WORD   .$.
6661  021614  104400                  MOVE    MEM,BREG                ;RESTORE BREG...
6662                                  MICPC=MICPC+1
6663  021614  000054                  .WORD   .$.
6664  021616  040620                  SBR     12$                     ;LOOP ON ERROR...
6665                                  MICPC=MICPC+1
6666  021616  000055                  .WORD   .$.
6667  021620  100435          13$:    CALL    SCP1                    ;IS LOOP DATA SET???
```

```
021620                          MOVE    # <MICPC+3>,BREG
000056
021620  000460                  MICPC=MICPC+1
021622                          .WORD   .$.
000057                          SBR     SCP1
021622  104427                  MICPC=MICPC+1
021624                          .WORD   .$.
000060                          MOVE    MEM,BREG                ;RESTORE BREG...
021624  040620                  MICPC=MICPC+1
021626                          .WORD   .$.
000061                          SBR     12$                     ;LOOP ON DATA.
021626  100435                  MICPC=MICPC+1
021630                          .WORD   .$.
000062                          MOVE    MEM,BREG                ;RESTORE BREG...
021630  040620                  MICPC=MICPC+1
021632                          .WORD   .$.
000063                          SHFRT                           ;SET THE NEXT BIT...
021632  061620                  MICPC=MICPC+1
021634                          .WORD   .SBR!.SELB!.DBRSH
000064                          BR7     12$                     ;BRANCH IF NOT DONE...
021634  103435                  MICPC=MICPC+1
021636                          .WORD   .$.
                        14$:
021636                          CALL    SCPE
021636                          MOVE    # <MICPC+3>,BREG
000065                          MICPC=MICPC+1
021636  000467                  .WORD   .$.
021640                          SBR     SCPE
000066                          MICPC=MICPC+1
021640  104454                  .WORD   .$.
021642                          SBR     21$
000067                          MICPC=MICPC+1
021642  100400                  .WORD   .$.
021644                  SSPTS1  13
021644                  SXZ


                                ;*************************** TEST 27 ****************************
                                ;* SCRATCH PAD TEST FOR SP13
                                ;* FLOAT A 1 THROUGH SCRATCH PAD 13
                                ;* FLOAT A 0 THROUGH SCRATCH PAD 13
021644                  SXZ
                                ;:************************************************************************
021644                  STSTN
                                ;  TEST 27
                                ;-------------
021644  012737  000027  001202  TST27:  MOV     #27,$TSTNM              ; LOAD THE NO. OF THIS TEST
021652  012737  022060  001442          MOV     #TST30,NEXT             ; POINT TO THE START OF NEXT TEST.
                                ;R1 CONTAINS BASE KMC11 ADDRESS
021660  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
021664  021700                  MCT27
021666  104022                  ERROR   22                      ;TIME OUT ERROR...
021670  012706  001200          MOV     #STACK,SP               ;RESET STACK...
021674  000177  157542          JMP     @NEXT                   ;GO TO NEXT TEST...
021700                  MCT27:
021700                  21$:
```

```
6724  021700                    MOVE    # 0,BREG         ;SET TO CLEAR SPAD 16...
6725          000000            MICPC=MICPC+1
6726  021700  000400            .WORD   .$.
6727  021702                    MOVE    BREG,SPAD <16>   ;FOR RETURN ADDRESS PURPOSES.
6728          000001            MICPC=MICPC+1
6729  021702  063236            .WORD   .$.
6730  021704              $SPFLT  1,1$,2$,3$,4$,1$
6731  021704              1$:     MOVE # 200,BREG ;START WITH BIT 7.
6732          000002            MICPC=MICPC+1
6733  021704  000600            .WORD   .$.
6734  021706              2$:     MOVE    BREG,SPAD <13>   ;LOAD THE SCRATCH PAD...
6735          000003            MICPC=MICPC+1
6736  021706  063233            .WORD   .$.
6737  021710                    MOVE    SPAD <13>,MEM    ;GET THE "FOUND"...
6738          000004            MICPC=MICPC+1
6739  021710  062613            .WORD   .$.
6740  021712                    MOVE    MEM,SPAD <0>             ;
6741          000005            MICPC=MICPC+1
6742  021712  043220            .WORD   .$.
6743  021714                    MOVE    BREG,MEM                 ;SAVE THE CONTENTS OF BREG...
6744          000006            MICPC=MICPC+1
6745  021714  062620            .WORD   .$.
6746  021716                    $IFEQ   BREG,SPAD <0>   3$       ;IF GOOD.. CONTINUE...
6747
6748
6749  021716                    SUB2C   SPAD <0>,BREG,NOP
6750          000007            MICPC=MICPC+1
6751  021716  060360            .WORD   .$.
6752  021720                    BZ      3$
6753          000010            MICPC=MICPC+1
6754  021720  101424            .WORD   .$.
6755  021722                    MOVE    BREG,OUT1 <CSR4>         ;ELSE, REPORT ERROR...
6756          000011            MICPC=MICPC+1
6757  021722  061224            .WORD   .$.
6758  021724                    MOVE    SPAD <0>,BREG            ;
6759          000012            MICPC=MICPC+1
6760  021724  060600            .WORD   .$.
6761  021726                    MOVE    BREG,OUT1 <CSR5>         ;BAD DATA...
6762          000013            MICPC=MICPC+1
6763  021726  061225            .WORD   .$.
6764  021730                    MOVE    # 4,BREG                 ;TYPE OF ERROR...
6765          000014            MICPC=MICPC+1
6766  021730  000404            .WORD   .$.
6767  021732                    MOVE    BREG,OUT1 <CSR3>         ;
6768          000015            MICPC=MICPC+1
6769  021732  061223            .WORD   .$.
6770  021734                    MOVE    # 13,BREG                ;LOAD ADDRESS.
6771          000016            MICPC=MICPC+1
6772  021734  000413            .WORD   .$.
6773  021736                    MOVE    BREG,OUT1 <CSR7>         ;
6774          000017            MICPC=MICPC+1
6775  021736  061227            .WORD   .$.
6776  021740                    CALL    ERROR                    ;DATA ERROR!!
6777  021740                    MOVE    # <MICPC+3>,BREG
6778          000020            MICPC=MICPC+1
6779  021740  000422            .WORD   .$.
```

```
6780  021742                    SBR     ERDR
6781          000021            MICPC=MICPC+1
6782  021742  104400            .WORD   .$.
6783  021744                    MOVE    MEM,BREG                ;RESTORE BREG...
6784          000022            MICPC=MICPC+1
6785  021744  040620            .WORD   .$.
6786  021746                    SBR     2$                      ;LOOP ON ERROR...
6787          000023            MICPC=MICPC+1
6788  021746  100403            .WORD   .$.
6789  021750              3$:   CALL    $CP1                    ;IS LOOP DATA SET???
6790  021750                    MOVE    @ <MICPC+3>,BREG
6791          000024            MICPC=MICPC+1
6792  021750  000426            .WORD   .$.
6793  021752                    SBR     $CP1
6794          000025            MICPC=MICPC+1
6795  021752  104427            .WORD   .$.
6796  021754                    MOVE    MEM,BREG                ;RESTORE BREG...
6797          000026            MICPC=MICPC+1
6798  021754  040620            .WORD   .$.
6799  021756                    SBR     2$                      ;LOOP ON DATA.
6800          000027            MICPC=MICPC+1
6801  021756  100403            .WORD   .$.
6802  021760                    MOVE    MEM,BREG                ;RESTORE BREG...
6803          000030            MICPC=MICPC+1
6804  021760  040620            .WORD   .$.
6805  021762                    SHFBRT                          ;SET THE NEXT BIT...
6806          000031            MICPC=MICPC+1
6807  021762  061620            .WORD   .SBR!.SELB!.DBRSH
6808  021764                    BB7     4$                      ;BRANCH IF DONE...
6809          000032            MICPC=MICPC+1
6810  021764  103434            .WORD   .$.
6811  021766                    SBR     2$                      ;CONTINUE...
6812          000033            MICPC=MICPC+1
6813  021766  100403            .WORD   .$.
6814  021770              4$:
6815  021770              $SPFLT  0,11$,12$,13$,14$,13
6816  021770              11$:    MOVE  @ 177,BREG ;START WITH BIT 7.
6817          000034            MICPC=MICPC+1
6818  021770  000577            .WORD   .$.
6819  021772              12$:    MOVE    BREG,SPAD <13>  ;LOAD THE SCRATCH PAD...
6820          000035            MICPC=MICPC+1
6821  021772  063233            .WORD   .$.
6822  021774                    MOVE    SPAD <13>,MEM   ;GET THE "FOUND"...
6823          000036            MICPC=MICPC+1
6824  021774  062613            .WORD   .$.
6825  021776                    MOVE    MEM,SPAD <0>            ;
6826          000037            MICPC=MICPC+1
6827  021776  043220            .WORD   .$.
6828  022000                    MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
6829          000040            MICPC=MICPC+1
6830  022000  062620            .WORD   .$.
6831  022002                    SIFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...
6832
6833
6834  022002                    SUB2C   SPAD <0>,BREG,NOP
6835          000041            MICPC=MICPC+1
```

```
                        .WORD   .$
                        BZ      13$
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    SPAD <0>,BREG           ;
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    # 4,BREG                ;TYPE OF ERROR...
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    BREG,OUT1 <CSR3>        ;
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    # 13,BREG               ;LOAD ADDRESS.
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    BREG,OUT1 <CSR7>        ;
                        MICPC=MICPC+1
                        .WORD   .$
                        CALL    ERROR                   ;DATA ERROR!!
                        MOVE    # <MICPC+3>,BREG
                        MICPC=MICPC+1
                        .WORD   .$
                        SBR     ERROR
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
                        .WORD   .$
                        SBR     12$                     ;LOOP ON ERROR...
                        MICPC=MICPC+1
                        .WORD   .$
                13$:    CALL    SCP1                    ;IS LOOP DATA SET???
                        MOVE    # <MICPC+3>,BREG
                        MICPC=MICPC+1
                        .WORD   .$
                        SBR     SCP1
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
                        .WORD   .$
                        SBR     12$                     ;LOOP ON DATA.
                        MICPC=MICPC+1
                        .WORD   .$
                        MOVE    MEM,BREG                ;RESTORE BREG...
                        MICPC=MICPC+1
                        .WORD   .$
                        SHFBRT                          ;SET THE NEXT BIT...
                        MICPC=MICPC+1
```

```
6892  022046  061620                        .WORD   .SBR!.SELB!.DBRSH
6893  022050                                BR7     12$                 ;BRANCH IF NOT DONE...
6894          000064                        MICPC=MICPC+1
6895  022050  103435                        .WORD   .$.
6896  022052                        14$:
6897  022052                                CALL    SCPE
6898  022052                                MOVE    # <MICPC+3),BREG
6899          000065                        MICPC=MICPC+1
6900  022062  000467                        .WORD   .$.
6901  022054                                $BR     SCPE
6902          000066                        MICPC=MICPC+1
6903  022054  104454                        .WORD   .$.
6904  022056                                $BR     21$
6905          000067                        MICPC=MICPC+1
6906  022056  100400                        .WORD   .$.
6907  022060                        $SPTS1  14
6908  022060                        $XZ
6909
6910
6911                                        ;*************************** TEST 30 ****************************
6912                                        ;* SCRATCH PAD TEST FOR SP14
6913                                        ;* FLOAT A 1 THROUGH SCRATCH PAD 14
6914                                        ;* FLOAT A 0 THROUGH SCRATCH PAD 14
6915  022060                        $XZ
6916                                        ;:***************************************************************
6917
6918  022060                        $TSTN
6919
6920                                        ;  TEST 30
6921                                        ;-------------
6922  022060  012737  000030  001202  TST30: MOV     #30,$TSTNM              ; LOAD THE NO. OF THIS TEST
6923  022066  012737  022274  001442         MOV     #TST31,NEXT             ; POINT TO THE START OF NEXT TEST.
6924                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
6925  022074  004737  035536                JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
6926  022100  022114                        MCT30
6927  022102  104022                        ERROR   22                      ;TIME OUT ERROR...
6928  022104  012706  001200                MOV     #STACK,SP               ;RESET STACK...
6929  022110  000177  157326                JMP     @NEXT                   ;GO TO NEXT TEST...
6930  022114                        MCT30:
6931  022114                        21$:
6932          000000                        MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16...
6933  022114  000400                        MICPC=MICPC+1
6934  022116                                .WORD   .$.
6935          000001                        MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
6936  022116  063236                        MICPC=MICPC+1
6937  022120                                .WORD   .$.
6938  022120                        $SPFLT  1,1$,2$,3$,4$,14
6939          000002                1$:     MOVE # 200,BREG ;START WITH BIT 7.
6940  022120  000600                        MICPC=MICPC+1
6941  022122                                .WORD   .$.
6942          000003                2$:     MOVE    BREG,SPAD <14>  ;LOAD THE SCRATCH PAD...
6943  022122  063234                        MICPC=MICPC+1
6944  022124                                .WORD   .$.
6945          000004                        MOVE    SPAD <14>,MEM   ;GET THE "FOUND"...
6946  022124  062614                        MICPC=MICPC+1
6947  022126                                .WORD   .$.
                                            MOVE    MEM,SPAD <0>            ;
```

```
                    000005                  MICPC=MICPC+1
022126  043220      022130                  .WORD   .$.
                    000006                  MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
022130  062620      022132                  MICPC=MICPC+1
                                            .WORD   .$.
                                            $IFEQ   BREG,SPAD <0>   3$      ;IF GOOD.. CONTINUE...

022132                                      SUB2C   SPAD <0>,BREG,NOP
                    000007                  MICPC=MICPC+1
022132  060360      022134                  .WORD   .$.
                    000010                  BZ      3$
022134  101424      022136                  MICPC=MICPC+1
                                            .WORD   .$.
                    000011                  MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
022136  061224      022140                  MICPC=MICPC+1
                                            .WORD   .$.
                    000012                  MOVE    SPAD <0>,BREG           ;
022140  060600      022142                  MICPC=MICPC+1
                                            .WORD   .$.
                    000013                  MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
022142  061225      022144                  MICPC=MICPC+1
                                            .WORD   .$.
                    000014                  MOVE    #4,BREG                 ;TYPE OF ERROR...
022144  000404      022146                  MICPC=MICPC+1
                                            .WORD   .$.
                    000015                  MOVE    BREG,OUT1 <CSR3>        ;
022146  061223      022150                  MICPC=MICPC+1
                                            .WORD   .$.
                    000016                  MOVE    #14,BREG                ;LOAD ADDRESS.
022150  000414      022152                  MICPC=MICPC+1
                                            .WORD   .$.
                    000017                  MOVE    BREG,OUT1 <CSR7>        ;
022152  061227      022154                  MICPC=MICPC+1
                                            .WORD   .$.
022154                                      CALL    ERROR                   ;DATA ERROR!!
                    000020                  MOVE    @<MICPC+3>,BREG
022154  000422      022156                  MICPC=MICPC+1
                                            .WORD   .$.
                    000021                  SBR     ERROR
022156  104400      022160                  MICPC=MICPC+1
                                            .WORD   .$.
                    000022                  MOVE    MEM,BREG                ;RESTORE BREG...
022160  040620      022162                  MICPC=MICPC+1
                                            .WORD   .$.
                    000023                  SBR     2$                      ;LOOP ON ERROR...
022162  100403      022164                  MICPC=MICPC+1
                                            .WORD   .$.
                                    3$:     CALL    SCP1                    ;IS LOOP DATA SET???
                    000024                  MOVE    @<MICPC+3>,BREG
022164  000426      022166                  MICPC=MICPC+1
                                            .WORD   .$.
                    000025                  SBR     SCP1
022166  104427      022170                  MICPC=MICPC+1
                                            .WORD   .$.
                                            MOVE    MEM,BREG                ;RESTORE BREG...
```

```
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     2$                      ;LOOP ON DATA.
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    MEM,BREG                ;RESTORE BREG...
                                MICPC=MICPC+1
                                .WORD   .$.
                                SHFRT                           ;SET THE NEXT BIT...
                                MICPC=MICPC+1
                                .WORD   .SBR!.SELB!.DBRSH
                                BR7     4$                      ;BRANCH IF DONE...
                                MICPC=MICPC+1
                                .WORD   .$.
                                SBR     2$                      ;CONTINUE...
                                MICPC=MICPC+1
                                .WORD   .$.
                        4$:
                        $SPFLT  0,11$,12$,13$,14$,14
                        11$:    MOVE # 177,BREG ;START WITH BIT 7.
                                MICPC=MICPC+1
                                .WORD   .$.
                        12$:    MOVE    BREG,SPAD (14)  ;LOAD THE SCRATCH PAD...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    SPAD (14),MEM   ;GET THE "FOUND"...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    MEM,SPAD (0)            ;
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
                                MICPC=MICPC+1
                                .WORD   .$.
                                $IFEQ   BREG,SPAD (0)   13$     ;IF GOOD.. CONTINUE...

                                SUB2C   SPAD (0),BREG,NOP
                                MICPC=MICPC+1
                                .WORD   .$.
                                BZ      13$
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,OUT1 (CSR4)        ;ELSE, REPORT ERROR...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    SPAD (0),BREG           ;
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,OUT1 (CSR5)        ;BAD DATA...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    # 4,BREG                ;TYPE OF ERROR...
                                MICPC=MICPC+1
                                .WORD   .$.
                                MOVE    BREG,OUT1 (CSR3)        ;
```

```
7060  000047
7061  022232  061223
7062  022234
7063  000050
7064  022234  000414
7065  022236
7066  000051
7067  022236  061227
7068  022240
7069  022240
7070  000052
7071  022240  000454
7072  022242
7073  000053
7074  022242  104400
7075  022244
7076  000054
7077  022244  040620
7078  022246
7079  000055
7080  022246  100435
7081  022250
7082  022250
7083  000056
7084  022250  000460
7085  022252
7086  000057
7087  022252  104427
7088  022254
7089  000060
7090  022254  040620
7091  022256
7092  000061
7093  022256  100435
7094  022260
7095  000062
7096  022260  040620
7097  022262
7098  000063
7099  022262  061620
7100  022264
7101  000064
7102  022264  103435
7103  022266
7104  022266
7105  022266
7106  000065
7107  022266  000467
7108  022270
7109  000066
7110  022270  104454
7111  022272
7112  000067
7113  022272  100400
7114  022274
7115  022274
```

```
                MICPC=MICPC+1
                .WORD   .$
                MOVE    # 14,BREG               ;LOAD ADDRESS.
                MICPC=MICPC+1
                .WORD   .$
                MOVE    BREG,OUT1 <CSR7>        ;
                MICPC=MICPC+1
                .WORD   .$
                CALL    ERROR                   ;DATA ERROR!!
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$
                SBR     ERROR
                MICPC=MICPC+1
                .WORD   .$
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$
                SBR     12$                     ;LOOP ON ERROR...
                MICPC=MICPC+1
                .WORD   .$
13$:            CALL    SCP1                    ;IS LOOP DATA SET???
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$
                SBR     12$                     ;LOOP ON DATA.
                MICPC=MICPC+1
                .WORD   .$
                MOVE    MEM,BREG                ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$
                SHFBRT                          ;SET THE NEXT BIT...
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                BR7     12$                     ;BRANCH IF NOT DONE...
                MICPC=MICPC+1
                .WORD   .$
14$:
                CALL    SCPE
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$
                SBR     SCPE
                MICPC=MICPC+1
                .WORD   .$
                SBR     21$
                MICPC=MICPC+1
                .WORD   .$
$SPTS1  15
$XZ
```

```
                                        ;************************** TEST 31 **************************
                                        ;* SCRATCH PAD TEST FOR SP15
                                        ;* FLOAT A 1 THROUGH SCRATCH PAD 15
                                        ;* FLOAT A 0 THROUGH SCRATCH PAD 15
022274                          $XZ
                                        ;:*************************************************************
022274                          $TSTN
                                ;  TEST 31
                                ;-------------
022274 012737 000031 001202 TST31:  MOV     #31,$TSTNM              ; LOAD THE NO. OF THIS TEST
022302 012737 022510 001442         MOV     #TST32,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
022310 004737 035536                JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
022314 022330                       MCT31
022316 104022                       ERROR   22                      ;TIME OUT ERROR...
022320 012706 001200                MOV     #STACK,SP               ;RESET STACK...
022324 000177 157112                JMP     @NEXT                   ;GO TO NEXT TEST...
022330                          MCT31:
022330                          21$:
022330                              MOVE    # 0,BREG            ;SET TO CLEAR SPAD 16...
000000                              MICPC=MICPC+1
022332 000400                       .WORD   .$.
022332                              MOVE    BREG,SPAD (16)      ;FOR RETURN ADDRESS PURPOSES.
000001                              MICPC=MICPC+1
022334 063236                       .WORD   .$.
022334                          $SPFLT  1,15,25,35,45,15
022334                          15:     MOVE # 200,BREG ;START WITH BIT 7.
000002                              MICPC=MICPC+1
022336 000600                       .WORD   .$.
022336                          25:     MOVE    BREG,SPAD (15)  ;LOAD THE SCRATCH PAD...
000003                              MICPC=MICPC+1
022340 063235                       .WORD   .$.
022340                              MOVE    SPAD (15),MEM   ;GET THE "FOUND"...
000004                              MICPC=MICPC+1
022342 062615                       .WORD   .$.
022342                              MOVE    MEM,SPAD (0)            ;
000005                              MICPC=MICPC+1
022344 043220                       .WORD   .$.
022344                              MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
000006                              MICPC=MICPC+1
022346 062620                       .WORD   .$.
022346                              $IFEQ   BREG,SPAD (0)   3$      ;IF GOOD.. CONTINUE...

022346                              SUB2C   SPAD (0),BREG,NOP
000007                              MICPC=MICPC+1
022346 060360                       .WORD   .$.
022350                              BZ      3$
000010                              MICPC=MICPC+1
022350 101424                       .WORD   .$.
022352                              MOVE    BREG,OUT1 (CSR+4)       ;ELSE, REPORT ERROR...
000011                              MICPC=MICPC+1
022352 061224                       .WORD   .$.
```

```
7172  022354                    MOVE    SPAD <0>,BREG           ;
7173        000012              MICPC=MICPC+1
7174  022354 060600             .WORD   .S.
7175  022356                    MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
7176        000013              MICPC=MICPC+1
7177  022356 061225             .WORD   .S.
7178  022360                    MOVE    # 4,BREG                ;TYPE OF ERROR...
7179        000014              MICPC=MICPC+1
7180  022360 000404             .WORD   .S.
7181  022362                    MOVE    BREG,OUT1 <CSR3>        ;
7182        000015              MICPC=MICPC+1
7183  022362 061223             .WORD   .S.
7184  022364                    MOVE    # 15,BREG               ;LOAD ADDRESS.
7185        000016              MICPC=MICPC+1
7186  022364 000415             .WORD   .S.
7187  022366                    MOVE    BREG,OUT1 <CSR7>        ;
7188        000017              MICPC=MICPC+1
7189  022366 061227             .WORD   .S.
7190  022370                    CALL    ERROR                   ;DATA ERROR!!
7191  022370                    MOVE    # <MICPC+3>,BREG
7192        000020              MICPC=MICPC+1
7193  022370 000422             .WORD   .S.
7194  022372                    SBR     ERROR
7195        000021              MICPC=MICPC+1
7196  022372 104400             .WORD   .S.
7197  022374                    MOVE    MEM,BREG                ;RESTORE BREG...
7198        000022              MICPC=MICPC+1
7199  022374 040620             .WORD   .S.
7200  022376                    SBR     2S                      ;LOOP ON ERROR...
7201        000023              MICPC=MICPC+1
7202  022376 100403             .WORD   .S.
7203  022400                3S: CALL    SCP1                    ;IS LOOP DATA SET???
7204  022400                    MOVE    # <MICPC+3>,BREG
7205        000024              MICPC=MICPC+1
7206  022400 000426             .WORD   .S.
7207  022402                    SBR     SCP1
7208        000025              MICPC=MICPC+1
7209  022402 104427             .WORD   .S.
7210  022404                    MOVE    MEM,BREG                ;RESTORE BREG...
7211        000026              MICPC=MICPC+1
7212  022404 040620             .WORD   .S.
7213  022406                    SBR     2S                      ;LOOP ON DATA.
7214        000027              MICPC=MICPC+1
7215  022406 100403             .WORD   .S.
7216  022410                    MOVE    MEM,BREG                ;RESTORE BREG...
7217        000030              MICPC=MICPC+1
7218  022410 040620             .WORD   .S.
7219  022412                    SHFBRT                          ;SET THE NEXT BIT...
7220        000031              MICPC=MICPC+1
7221  022412 061620             .WORD   .SBR!.SELB!.DRSH
7222  022414                    BR7     4S                      ;BRANCH IF DONE...
7223        000032              MICPC=MICPC+1
7224  022414 103434             .WORD   .S.
7225  022416                    SBR     2S                      ;CONTINUE...
7226        000033              MICPC=MICPC+1
7227  022416 100403             .WORD   .S.
```

```
022420
022420
022420
                000034
022420          000577
022422
                000035
022422          063235
022424
                000036
022424          062615
022426
                000037
022426          043220
022430
                000040
022430          062620
022432

022432
                000041
022432          060360
022434
                000042
022434          101456
022436
                000043
022436          061224
022440
                000044
022440          060600
022442
                000045
022442          061225
022444
                000046
022444          000404
022446
                000047
022446          061223
022450
                000050
022450          000415
022452
                000051
022452          061227
022454
022454
                000052
022454          000454
022456
                000053
022456          104400
022460
                000054
```

```
4$:
SSPFLT  0,11$,12$,13$,14$,15
11$:    MOVE  # 177,BREG ;START WITH BIT 7.
        MICPC=MICPC+1
        .WORD   .$.
12$:    MOVE    BREG,SPAD <15>  ;LOAD THE SCRATCH PAD...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <15>,MEM   ;GET THE "FOUND"...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,SPAD <0>            ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,MEM                ;SAVE THE CONTENTS OF BREG...
        MICPC=MICPC+1
        .WORD   .$.
        SIFEQ   BREG,SPAD <0>   13$     ;IF GOOD.. CONTINUE...


        SUB2C   SPAD <0>,BREG,NOP
        MICPC=MICPC+1
        .WORD   .$.
        BZ      13$
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 <CSR4>        ;ELSE, REPORT ERROR...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    SPAD <0>,BREG           ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 <CSR5>        ;BAD DATA...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 4,BREG                ;TYPE OF ERROR...
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 <CSR3>        ;
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    # 15,BREG                ;LOAD ADDRESS.
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    BREG,OUT1 <CSR7>        ;
        MICPC=MICPC+1
        .WORD   .$.
        CALL    EROR                    ;DATA ERROR!!
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .$.
        SBR     EROR
        MICPC=MICPC+1
        .WORD   .$.
        MOVE    MEM,BREG                ;RESTORE BREG...
        MICPC=MICPC+1
```

```
7284  022460  040620                    .WORD   .$.
7285  022462                            $BR     12$                     ;LOOP ON ERROR...
7286          000055                    MICPC=MICPC+1
7287  022462  100435                    .WORD   .$.
7288  022464                    13$:    CALL    SCP1                    ;IS LOOP DATA SET???
7289  022464                            MOVE    @ <MICPC+3>,BREG
7290          000056                    MICPC=MICPC+1
7291  022464  000460                    .WORD   .$.
7292  022466                            $BR     .+2
7293          000057                    MICPC=MICPC+1
7294  022466  104427                    .WORD   .$.
7295  022470                            MOVE    MEM,BREG                ;RESTORE BREG...
7296          000060                    MICPC=MICPC+1
7297  022470  040620                    .WORD   .$.
7298  022472                            $BR     12$                     ;LOOP ON DATA.
7299          000061                    MICPC=MICPC+1
7300  022472  100435                    .WORD   .$.
7301  022474                            MOVE    MEM,BREG                ;RESTORE BREG...
7302          000062                    MICPC=MICPC+1
7303  022474  040620                    .WORD   .$.
7304  022476                            SHFRT                           ;SET THE NEXT BIT...
7305          000063                    MICPC=MICPC+1
7306  022476  061620                    .WORD   .SBR!.SELB!.DBRSH
7307  022500                            BR7     12$                     ;BRANCH IF NOT DONE...
7308          000064                    MICPC=MICPC+1
7309  022500  103435                    .WORD   .$.
7310  022502                    14$:
7311  022502                            CALL    SCPE
7312  022502                            MOVE    @ <MICPC+3>,BREG
7313          000065                    MICPC=MICPC+1
7314  022502  000467                    .WORD   .$.
7315  022504                            $BR     SCPE
7316          000066                    MICPC=MICPC+1
7317  022504  104454                    .WORD   .$.
7318  022506                            $BR     21$
7319          000067                    MICPC=MICPC+1
7320  022506  100400                    .WORD   .$.
7321  022510                    SNPR1
7322  022510                    SXZ
7323
7324
7325                                    ;*************************** TEST 32 ****************************
7326                                    ;* NPR TEST
7327                                    ;* TEST OF DATI, 1 WORD FROM 11 MEMORY TO UPROC
7328  022510                    SXZ     ;;************************************************************************
7329
7330
7331  022510                    STSTN
7332
7333                                    ;  TEST 32
7334                                    ;-------------
7335  022510  012737  000032  001202  TST32:  MOV     #32,$TSTNM              ; LOAD THE NO. OF THIS TEST
7336  022516  012737  022704  001442          MOV     #TST33,NEXT             ; POINT TO THE START OF NEXT TEST.
7337                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
7338  022524  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
7339  022530  022544                          MCT32
7340  022532  104022                          ERROR   22                      ;TIME OUT ERROR...
```

```
7340  022534  012706  001200          MOV     #STACK,SP               ;RESET STACK...
7341  022540  000177  156676          JMP     @NEXT                   ;GO TO NEXT TEST...
7342  022544                  MCT32:
7343  022544                  1$:     MOVE    # 0,BREG
7344                                  MICPC=MICPC+1                   ;SET TO CLEAR NPR IBUS REGISTERS.
7345  022544  000000                   .WORD   .$.
7346  022546  000400                   MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS PURPOSE...
7347                                  MICPC=MICPC+1
7348  022546  000001                   .WORD   .$.
7349  022550  063236                   MOVE    BREG,OUT0 (0)           ;CLEAR IBUS+IN DATA+LB.
7350                                  MICPC=MICPC+1
7351  022550  000002                   .WORD   .$.
7352  022552  062220                   MOVE    BREG,OUT0 (1)           ;CLEAR IBUS+IN DATA+HB.
7353                                  MICPC=MICPC+1
7354  022552  000003                   .WORD   .$.
7355  022554  062221                   MOVE    BREG,OUT0 (4)           ;CLEAR INBA+LB.
7356                                  MICPC=MICPC+1
7357  022554  000004                   .WORD   .$.
7358  022556  062224                   MOVE    BREG,OUT0 (5)           ;CLEAR INBA+HB.
7359                                  MICPC=MICPC+1
7360  022556  000005                   .WORD   .$.
7361  022560  062225                   MOVE    BREG,OUT1 (CSR10)       ;CLEAR EA+BITS.
7362                                  MICPC=MICPC+1
7363  022560  000006                   .WORD   .$.
7364  022562  061230                   MOVE    # 125,BREG              ;SET GOOD DATA.
7365                                  MICPC=MICPC+1
7366  022562  000007                   .WORD   .$.
7367  022564  000525                   MOVE    BREG,SPAD (3)   ; " " " .
7368                                  MICPC=MICPC+1
7369  022564  000010                   .WORD   .$.
7370  022566  063223                   SHFRT
7371                                  MICPC=MICPC+1
7372  022566  000011                   .WORD   .SBR!.SELB!.DBRSH
7373  022570  061620                   MOVE    BREG,SPAD (4)           ; " " " .
7374                                  MICPC=MICPC+1
7375  022570  000012                   .WORD   .$.
7376  022572  063224                   MOVE    # (NPDAT&377),BREG      ;SET ADDRESS.
7377                                  MICPC=MICPC+1
7378  022572  000013                   .WORD   .$.
7379  022574  000702                   MOVE    BREG,SPAD (1)           ; " "  .
7380                                  MICPC=MICPC+1
7381  022574  000014                   .WORD   .$.
7382  022576  063221                   MOVE    # (NPDAT&177400/400),BREG       ;
7383                                  MICPC=MICPC+1
7384  022576  000015                   .WORD   .$.
7385  022600  000445                   MOVE    BREG,SPAD (2)           ;
7386                                  MICPC=MICPC+1
7387  022600  000016                   .WORD   .$.
7388  022602  063222          2$:     MOVE    SPAD (1),BREG           ;SET INBA+ADDRESS
7389                                  MICPC=MICPC+1
7390  022602  000017                   .WORD   .$.
7391  022604  060601                   MOVE    BREG,OUT0 (4)           ;
7392                                  MICPC=MICPC+1
7393  022604  000020                   .WORD   .$.
7394  022606  062224                   MOVE    SPAD (2),BREG           ;SET INBA+ADDRESS
7395                                  MICPC=MICPC+1
                          000021
```

```
7396  022606  060602                    .WORD   .S.
7397  022610                            MOVE    BREG,OUT0 <5>           ;
7398          000022                    MICPC=MICPC+1
7399  022610  062225                    .WORD   .S.
7400  022612                            MOVE    # 001,BREG              ;SET NPR BIT.
7401          000023                    MICPC=MICPC+1
7402  022612  000401                    .WORD   .S.
7403  022614                            MOVE    BREG,OUT1 <CSR10>               ;SET NPR RQ BIT WITH IN NPR & WRD.XFR
7404          000024                    MICPC=MICPC+1
7405  022614  061230                    .WORD   .S.
7406  022616                    3$:     MOVE    INP1 <CSR10>,BREG               ;IS NPR DONE???
7407          000025                    MICPC=MICPC+1
7408  022616  120600                    .WORD   .S.
7409  022620                            BBD     3$                      ;NO, SPIN ON IT??
7410          000026                    MICPC=MICPC+1
7411  022620  102025                    .WORD   .S.
7412  022622                            MOVE    INP0 <0>,BREG           ;GET THE DATA LB.
7413          000027                    MICPC=MICPC+1
7414  022622  020400                    .WORD   .S.
7415  022624                            $IFEQ   BREG,SPAD <3>   4$      ;IF GOOD, CHECK NEXT BYTE.
7416
7417
7418  022624                            SUB2C   SPAD <3>,BREG,NOP
7419          000030                    MICPC=MICPC+1
7420  022624  060363                    .WORD   .S.
7421  022626                            BZ      4$
7422          000031                    MICPC=MICPC+1
7423  022626  101442                    .WORD   .S.
7424  022630                            MOVE    BREG,OUT1 <CSR5>        ;MOVE GOOD DATA...
7425          000032                    MICPC=MICPC+1
7426  022630  061225                    .WORD   .S.
7427  022632                            MOVE    SPAD <3>,BREG           ;MOVE BAD DATA...
7428          000033                    MICPC=MICPC+1
7429  022632  060603                    .WORD   .S.
7430  022634                            MOVE    BREG,OUT1 <CSR4>                ;
7431          000034                    MICPC=MICPC+1
7432  022634  061224                    .WORD   .S.
7433  022636                            MOVE    # 11,BREG               ;SET THE TYPE OF ERROR...
7434          000035                    MICPC=MICPC+1
7435  022636  000411                    .WORD   .S.
7436  022640                            MOVE    BREG,OUT1 <CSR3>        ;
7437          000036                    MICPC=MICPC+1
7438  022640  061223                    .WORD   .S.
7439  022642                            CALL    ERROR1                  ;REPORT ERROR...
7440  022642                            MOVE    @ <MICPC+3>,BREG
7441          000037                    MICPC=MICPC+1
7442  022642  000441                    .WORD   .S.
7443  022644                            SBR     ERROR1
7444          000040                    MICPC=MICPC+1
7445  022644  104401                    .WORD   .S.
7446  022646                            SBR     1$                      ;
7447          000041                    MICPC=MICPC+1
7448  022646  100400                    .WORD   .S.
7449  022650                    4$:     MOVE    INP0 <1>,BREG           ;GET THE NPR DATA MB.
7450          000042                    MICPC=MICPC+1
7451  022650  020420                    .WORD   .S.
```

```
7452 022652                        SIFEQ   BREG,SPAD <4>   5$      ;IF GOOD, BRANCH.
7453
7454
7455 022652                        SUB2C   SPAD <4>,BREG,NOP
7456 000043                        MICPC=MICPC+1
7457 022652 060364                 .WORD   .$.
7458 022654                        BZ      5$
7459 000044                        MICPC=MICPC+1
7460 022654 101454                 .WORD   .$.
7461 022656                        MOVE    BREG,OUT1 <CSR5>        ;GOOD DATA.
7462 000045                        MICPC=MICPC+1
7463 022656 061225                 .WORD   .$.
7464 022660                        MOVE    SPAD <4>,OUT1 <CSR4>    ;BAD DATA.
7465 000046                        MICPC=MICPC+1
7466 022660 061204                 .WORD   .$.
7467 022662                        MOVE    # 11,MEM                ;TYPE OF ERROR...
7468 000047                        MICPC=MICPC+1
7469 022662 002411                 .WORD   .$.
7470 022664                        MOVE    MEM,OUT1 <CSR3>         ;
7471 000050                        MICPC=MICPC+1
7472 022664 041223                 .WORD   .$.
7473 022666                        CALL    EROR1                   ;DATA ERROR...
7474 022666                        MOVE    @ <MICPC+3>,BREG
7475 000051                        MICPC=MICPC+1
7476 022666 000453                 .WORD   .$.
7477 022670                        SBR     EROR1
7478 000052                        MICPC=MICPC+1
7479 022670 104401                 .WORD   .$.
7480 022672                        SBR     1$                      ;LOOP ON ERROR...
7481 000053                        MICPC=MICPC+1
7482 022672 100400                 .WORD   .$.
7483 022674                 5$:    CALL    SCPE                    ;SCOPE THE TEST...
7484 022674                        MOVE    @ <MICPC+3>,BREG
7485 000054                        MICPC=MICPC+1
7486 022674 000456                 .WORD   .$.
7487 022676                        SBR     SCPE
7488 000055                        MICPC=MICPC+1
7489 022676 104454                 .WORD   .$.
7490 022700                        SBR     1$                      ;DO THE NEXT ITERATION...
7491 000056                        MICPC=MICPC+1
7492 022700 100400                 .WORD   .$.
7493 022702   125   252    NPDAT:  .BYTE   125,252                 ;
7494 022704                 SNPR2   21,WORD,1
7495 022704                 SXZ
7496
7497
7498                                ;*************************** TEST 33 ****************************
7499                                ;* NPR TEST
7500                                ;* TEST OF DATO, 1 WORD FROM UPROC TO PDP11 MEMORY.
7501                                ;* THEN DATI OF THAT WORD AND CHECK....
7502 022704                 SXZ
7503                                ;:************************************************************************
7504
7505 022704                 STSTN
7506                                ;  TEST 33
7507                                ;--------------
```

```
7508  022704  012737  000033  001202  TST33:  MOV     #33,$TSTNM              ; LOAD THE NO. OF THIS TEST
7509  022712  012737  023154  001442          MOV     #TST34,NEXT             ; POINT TO THE START OF NEXT TEST.
7510                                    ;R1 CONTAINS BASE KMC11 ADDRESS
7511  022720  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
7512  022724  022740                          MCT33
7513  022726  104022                          ERROR   22                      ;TIME OUT ERROR...
7514  022730  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
7515  022734  000177  156502                  JMP     @NEXT                   ;GO TO NEXT TEST...
7516  022740                          MCT33:
7517  022740                          1$:     MOVE    # 0,BREG                ;SET TO CLEAR NPR IBUS-REGISTERS.
7518          000000                          MICPC=MICPC+1
7519  022740  000400                          .WORD   .$.
7520  022742                                  MOVE    BREG,SPAD <16>          ;FO RETURN ADDRESS PURPOSE...
7521          000001                          MICPC=MICPC+1
7522  022742  063236                          .WORD   .$.
7523  022744                                  MOVE    BREG,OUT0 <0>           ;CLEAR IBUS-IN DATA-LB.
7524          000002                          MICPC=MICPC+1
7525  022744  062220                          .WORD   .$.
7526  022746                                  MOVE    BREG,OUT0 <1>           ;CLEAR IBUS-IN DATA-HB.
7527          000003                          MICPC=MICPC+1
7528  022746  062221                          .WORD   .$.
7529  022750                                  MOVE    BREG,OUT0 <2>           ;CLEAR IBUS-OUT DATA-LB.
7530          000004                          MICPC=MICPC+1
7531  022750  062222                          .WORD   .$.
7532  022752                                  MOVE    BREG,OUT0 <3>           ;CLEAR IBUS-OUT DATA-HB.
7533          000005                          MICPC=MICPC+1
7534  022752  062223                          .WORD   .$.
7535  022754                                  MOVE    BREG,OUT0 <4>           ;CLEAR IBUS-IN BA-LB.
7536          000006                          MICPC=MICPC+1
7537  022754  062224                          .WORD   .$.
7538  022756                                  MOVE    BREG,OUT0 <5>           ;CLEAR IBUS-IN BA-HB.
7539          000007                          MICPC=MICPC+1
7540  022756  062225                          .WORD   .$.
7541  022760                                  MOVE    BREG,OUT0 <6>           ;CLEAR IBUS-OUT BA-LB.
7542          000010                          MICPC=MICPC+1
7543  022760  062226                          .WORD   .$.
7544  022762                                  MOVE    BREG,OUT0 <7>           ;CLEAR IBUS-OUT BA-HB.
7545          000011                          MICPC=MICPC+1
7546  022762  062227                          .WORD   .$.
7547  022764                                  MOVE    BREG,OUT1 <CSR10>       ;CLEAR IN-BA EABITS.
7548          000012                          MICPC=MICPC+1
7549  022764  061230                          .WORD   .$.
7550  022766                                  MOVE    INP1 <CSR11>,SPAD <0>   ;CLEAR OUT-BA EA-BITS.
7551          000013                          MICPC=MICPC+1
7552  022766  123220                          .WORD   .$.
7553  022770                                  OR      SPAD <0>,BREG,OUT1 <CSR11>      ;CLEAR OUT-BA EA-BITS.
7554          000014                          MICPC=MICPC+1
7555  022770  061311                          .WORD   .$.
7556  022772                                  MOVE    # 125,BREG              ; GOOD DATA.
7557          000015                          MICPC=MICPC+1
7558  022772  000525                          .WORD   .$.
7559  022774                                  MOVE    BREG,SPAD <3>           ;
7560          000016                          MICPC=MICPC+1
7561  022774  063223                          .WORD   .$.
7562  022776                                  SHFBRT                          ;
7563          000017                          MICPC=MICPC+1
```

```
7564  022776  061620                    .WORD   .SBR!.SELB!.DBRSH
7565  023000                            MOVE    BREG,SPAD <4>           ;
7566          000020                    MICPC=MICPC+1
7567  023000  063224                    .WORD   .S.
7568  023002                            MOVE    # <NPRDTI&377>,BREG     ;SET NPR ADDRESS.
7569          000021                    MICPC=MICPC+1
7570  023002  000552                    .WORD   .S.
7571  023004                            MOVE    BREG,SPAD <1>           ;
7572          000022                    MICPC=MICPC+1
7573  023004  063221                    .WORD   .S.
7574  023006                            MOVE    # <NPRDTI&177400/400>,BREG      ;
7575          000023                    MICPC=MICPC+1
7576  023006  000446                    .WORD   .S.
7577  023010                            MOVE    BREG,SPAD <2>           ;
7578          000024                    MICPC=MICPC+1
7579  023010  063222                    .WORD   .S.
7580  023012                    25:     MOVE    SPAD <1>,BREG           ;SET OUT+BA+ADDRESS.
7581          000025                    MICPC=MICPC+1
7582  023012  060601                    .WORD   .S.
7583  023014                            MOVE    BREG,OUT0 <6>           ; "  "  "  "  .
7584          000026                    MICPC=MICPC+1
7585  023014  062226                    .WORD   .S.
7586  023016                            MOVE    SPAD <2>,BREG           ;SET OUT+BA+ADDRESS.
7587          000027                    MICPC=MICPC+1
7588  023016  060602                    .WORD   .S.
7589  023020                            MOVE    BREG,OUT0 <7>           ; "  "  "  "  .
7590          000030                    MICPC=MICPC+1
7591  023020  062227                    .WORD   .S.
7592  023022                            MOVE    SPAD <3>,BREG           ;SET OUT+DATA.
7593          000031                    MICPC=MICPC+1
7594  023022  060603                    .WORD   .S.
7595  023024                            MOVE    BREG,OUT0 <2>           ; "  "  "  .
7596          000032                    MICPC=MICPC+1
7597  023024  062222                    .WORD   .S.
7598  023026                            MOVE    SPAD <4>,BREG           ; "  "  "  .
7599          000033                    MICPC=MICPC+1
7600  023026  060604                    .WORD   .S.
7601  023030                            MOVE    BREG,OUT0 <3>           ; "  "  "  .
7602          000034                    MICPC=MICPC+1
7603  023030  062223                    .WORD   .S.
7604  023032                            MOVE    # 021,BREG              ;SET NPR+OUT + NPR+RQ.
7605          000035                    MICPC=MICPC+1
7606  023032  000421                    .WORD   .S.
7607  023034                            MOVE    INP1 <CSR10>,SPAD <0>   ;GET THE CSR10 REGISTER.
7608          000036                    MICPC=MICPC+1
7609  023034  123200                    .WORD   .S.
7610  023036                            OR      BREG,SPAD <0>,OUT1 <CSR10>;SET NPR+OUT & NPR+RQ & WRD+XFR.
7611          000037                    MICPC=MICPC+1
7612  023036  061310                    .WORD   .S.
7613  023040                    35:     MOVE    INP1 <CSR10>,BREG       ;CHECK IF NPR DONE...
7614          000040                    MICPC=MICPC+1
7615  023040  120600                    .WORD   .S.
7616  023042                            BBO     35                      ;NO, SPIN ON IT...
7617          000041                    MICPC=MICPC+1
7618  023042  102040                    .WORD   .S.
7619                            ;:
```

```
7620
7621                            ;;      NOW GET IT BACK BY DATI.
7622 023044                     ;;
7623          000042                    MOVE    SPAD <1>,BREG           ;SET IN+BA ADDRESS.
7624 023044   060601                    MICPC=MICPC+1
7625 023046                             .WORD   .$.
7626          000043                    MOVE    BREG,OUT0 <4>           ; "  "  "    "   .
7627 023046   062224                    MICPC=MICPC+1
7630 023050                             .WORD   .$.
7631          000044                    MOVE    SPAD <2>,BREG           ; "  "  "    "   .
7632 023050   060602                    MICPC=MICPC+1
7633 023052                             .WORD   .$.
7634          000045                    MOVE    BREG,OUT0 <5>           ; "  "  "    "   .
7635 023052   062225                    MICPC=MICPC+1
7636 023054                             .WORD   .$.
7637          000046                    MOVE    # 001,BREG              ;SET NPR+RQ.
7640 023054   000401                    MICPC=MICPC+1
7641 023056                             .WORD   .$.
7642          000047                    MOVE    BREG,OUT1 <CSR10>       ;SET NPR+IN & NPR+RQ & WRD+XFR.
7643 023056   061230                    MICPC=MICPC+1
7644 023060                             .WORD   .$.
7645          000050            4$:     MOVE    INP1 <CSR10>,BREG       ;IS NPR DONE??
7646 023060   120600                    MICPC=MICPC+1
7647 023062                             .WORD   .$.
7650          000051                    BBO     4$                      ;NO, SPIN ON IT...
7651 023062   102050                    MICPC=MICPC+1
7652 023064                             .WORD   .$.
7653          000052                    MOVE    INP0 <0>,BREG           ;GET IN DATA+LB.
7654 023064   020400                    MICPC=MICPC+1
7655 023066                             .WORD   .$.
7656          000053                    MOVE    INP0 <2>,SPAD <0>       ;BRET OUT+DATA+LB.
7657 023066   023040                    MICPC=MICPC+1
7660 023070                             .WORD   .$.
7661                                    SIFEQ   BREG,SPAD <0> 6$        ;BRANCH IF GOOD..
7662
7663 023070                             SUB2C   SPAD <0>,BREG,NOP
7664          000054                    MICPC=MICPC+1
7665 023070   060360                    .WORD   .$.
7666 023072                             BZ      6$
7667          000055                    MICPC=MICPC+1
7670 023072   101466                    .WORD   .$.
7671 023074                             MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA
7672          000056                    MICPC=MICPC+1
7673 023074   061224                    .WORD   .$.
7674 023076                             MOVE    SPAD <0>,BREG           ;BAD DATA.
7675          000057                    MICPC=MICPC+1
7676 023076   060600                    .WORD   .$.
7677 023100                             MOVE    BREG,OUT1 <CSR5>        ;
7700          000060                    MICPC=MICPC+1
7701 023100   061225                    .WORD   .$.
7702 023102                             MOVE    # 11,MEM                ;SET ERROR TYPE...
7703          000061                    MICPC=MICPC+1
7704 023102   002411                    .WORD   .$.
7705 023104                             MOVE    MEM,OUT1 <CSR3>
7706          000062                    MICPC=MICPC+1
7707 023104   041223                    .WORD   .$.
```

```
023106                          CALL    ERORI
023106                          MOVE    @ <MICPC+3>,BREG ;DATA ERROR...
000063                          MICPC=MICPC+1
023106  000465                  .WORD   .$.
023110                          SBR     ERORI
000064                          MICPC=MICPC+1
023110  104401                  .WORD   .$.
023112                          SBR     .$              ;LOOP ON ERROR...
000065                          MICPC=MICPC+1
023112  100400                  .WORD   .$.
023114                  6$:     MOVE    INPO <1>,BREG   ;GET IN DATA + HB....
000066                          MICPC=MICPC+1
023114  020420                  .WORD   .$.
023116                          MOVE    INPO <3>,SPAD <0>       ;GET OUT+DATA+HB....
000067                          MICPC=MICPC+1
023116  023060                  .WORD   .$.
023120                          SIFEQ   BREG,SPAD <0> 7$        ;BRANCH IF GOOD...

023120                          SUB2C   SPAD <0>,BREG,NOP
000070                          MICPC=MICPC+1
023120  060360                  .WORD   .$.
023122                          BZ      7$
000071                          MICPC=MICPC+1
023122  101502                  .WORD   .$.
023124                          MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA.
000072                          MICPC=MICPC+1
023124  061224                  .WORD   .$.
023126                          MOVE    SPAD <0>,BREG           ;BAD DATA.
000073                          MICPC=MICPC+1
023126  060600                  .WORD   .$.
023130                          MOVE    BREG,OUT1 <CSR5>        ;
000074                          MICPC=MICPC+1
023130  061225                  .WORD   .$.
023132                          MOVE    @ 11,MEM                ;SET ERROR TYPE...
000075                          MICPC=MICPC+1
023132  002411                  .WORD   .$.
023134                          MOVE    MEM,OUT1 <CSR3>         ;
000076                          MICPC=MICPC+1
023134  041223                  .WORD   .$.
023136                          CALL    ERORI           ;DATA ERROR...
023136                          MOVE    @ <MICPC+3>,BREG
000077                          MICPC=MICPC+1
023136  000501                  .WORD   .$.
023140                          SBR     ERORI
000100                          MICPC=MICPC+1
023140  104401                  .WORD   .$.
023142                          SBR     .$              ;LOOP ON ERROR...
000101                          MICPC=MICPC+1
023142  100400                  .WORD   .$.
023144                  7$:     CALL    SCPE            ;SCOPE THE TEST...
023144                          MOVE    @ <MICPC+3>,BREG
000102                          MICPC=MICPC+1
023144  000504                  .WORD   .$.
023146                          SBR     SCPE
000103                          MICPC=MICPC+1
```

```
023146  104454                          .WORD   .$.
023150                                  SBR     1$              ;YES, DO NEXT ITERATION.
        000104                          MICPC=MICPC+1
023150  100400                          .WORD   .$.
023152  000000          NPRDTI:         .WORD   0               ;
023154                  SNPR9   21
023154                  SXZ


                        ;************************** TEST 34 **************************
                        ;#NPR NON-EXISTENT MEMORY TEST
                        ;#DO A DATO TO A NON-EXISTENT ADDRESS.
                        ;#VERIFY THAT THE NON-EXISTENT BIT SET IN IBUS REG 11
023154                  SXZ
                        ;;**********************************************************************
023154                  STSTN
                        ;  TEST 34
                        ;-------------
023154  012737  000034  001202  TST34:  MOV     #34,$TSTNM              ; LOAD THE NO. OF THIS TEST
023162  012737  023322  001442          MOV     #TST35,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
023170  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
023174  023210                          MCT34
023176  104022                          ERROR   22              ;TIME OUT ERROR...
023200  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
023204  000177  156232                  JMP     @NEXT                   ;GO TO NEXT TEST...
023210                          MCT34:
023210                          1$:     MOVE    # 0,BREG                 ;SET TO CLEAR IBUS+REGISTERS.
        000000                          MICPC=MICPC+1
023210  000400                          .WORD   .$.
023212                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
        000001                          MICPC=MICPC+1
023212  063236                          .WORD   .$.
023214                                  MOVE    BREG,SPAD <1>   ;CLEAR DELAY COUNT...
        000002                          MICPC=MICPC+1
023214  063221                          .WORD   .$.
023216                                  MOVE    BREG,OUT0 <0>
        000003                          MICPC=MICPC+1
023216  062220                          .WORD   .$.
023220                                  MOVE    BREG,OUT0 <1>   ;
        000004                          MICPC=MICPC+1
023220  062221                          .WORD   .$.
023222                                  MOVE    BREG,OUT0 <2>   ;
        000005                          MICPC=MICPC+1
023222  062222                          .WORD   .$.
023224                                  MOVE    BREG,OUT0 <3>   ;
        000006                          MICPC=MICPC+1
023224  062223                          .WORD   .$.
023226                                  MOVE    BREG,OUT0 <4>   ;
        000007                          MICPC=MICPC+1
023226  062224                          .WORD   .$.
023230                                  MOVE    BREG,OUT0 <5>   ;
        000010                          MICPC=MICPC+1
023230  062225                          .WORD   .$.
023232                                  MOVE    BREG,OUT0 <6>   ;
```

```
7788                    000011          MICPC=MICPC+1
7789  023232  062226                    .WORD   .S.
7790  023234                            MOVE    BREG,OUT0 <7>   ;
7791                    000012          MICPC=MICPC+1
7792  023234  062227                    .WORD   .S.
7793  023236                            MOVE    BREG,OUT1 <CSR10>       ;
7794                    000013          MICPC=MICPC+1
7795  023236  061230                    .WORD   .S.
7796  023240                    2$:     MOVE    #320,BREG       ;NPR ADDRESS+LB.
7797                    000014          MICPC=MICPC+1
7798  023240  000720                    .WORD   .S.
7799  023242                     MOVE   BREG,OUT0 <6>   ;LOAD OUT+BA+LB.
7800                    000015          MICPC=MICPC+1
7801  023242  062226                    .WORD   .S.
7802  023244                            MOVE    #376,BREG       ;NPR ADDRESS+HB.
7803                    000016          MICPC=MICPC+1
7804  023244  000776                    .WORD   .S.
7805  023246                     MOVE   BREG,OUT0 <7>   ;LOAD OUT+BA+HB.
7806                    000017          MICPC=MICPC+1
7807  023246  062227                    .WORD   .S.
7808  023250                            MOVE    #14,MEM                 ;SET OUT+BA BITS 16 &17.
7809                    000020          MICPC=MICPC+1
7810  023250  002414                    .WORD   .S.
7811  023252                            MOVE    INP1 <CSR11>,SPAD <0>   ;SET OUT+BA 16+17
7812                    000021          MICPC=MICPC+1
7813  023252  123220                    .WORD   .S.
7814  023254                            OR      MEM,SPAD <0>,OUT1 <CSR11>       ;SET OUT+BA 16 &17.
7815                    000022          MICPC=MICPC+1
7816  023254  041311                    .WORD   .S.
7817  023256                            MOVE    #21,BREG        ;
7818                    000023          MICPC=MICPC+1
7819  023256  000421                    .WORD   .S.
7820  023260                            MOVE    BREG,OUT1 <CSR10>       ;SET NPR+RQ OUT+NPR,WRD+XFR...
7821                    000024          MICPC=MICPC+1
7822  023260  061230                    .WORD   .S.
7823  023262                    3$:     MOVE    INP1 <CSR11>,BREG       ;CHECK IF NON+EX+MEM SET...
7824                    000025          MICPC=MICPC+1
7825  023262  120620                    .WORD   .S.
7826  023264                            BRO     4$              ;YES, ITS WORKING.
7827                    000026          MICPC=MICPC+1
7828  023264  102037                    .WORD   .S.
7829  023266                            SINC    SPAD <1>        ;WAIT FOR ATLEAST 18USEC.
7830                    000027          MICPC=MICPC+1
7831  023266  063061            .WORD   .S.!.DSP
7832  023270                            BC      6$              ;ENOUGH REPORT ERROR...
7833                    000030          MICPC=MICPC+1
7834  023270  101032                    .WORD   .S.
7835  023272                            SBR     3$              ;CHECK AGAIN...
7836                    000031          MICPC=MICPC+1
7837  023272  100425                    .WORD   .S.
7838  023274                    6$:     MOVE    #13,MEM                 ;SET ERROR TYPE.
7839                    000032          MICPC=MICPC+1
7840  023274  002413                    .WORD   .S.
7841  023276                            MOVE    MEM,OUT1 <CSR3>         ;
7842                    000033          MICPC=MICPC+1
7843  023276  041223                    .WORD   .S.
```

```
023300                          CALL    EROR1           ;REPORT ERROR...
023300                          MOVE    # <MICPC+3>,BREG
000034                          MICPC=MICPC+1
023300  000436                  .WORD   .$.
023302                          SBR     EROR1
000035                          MICPC=MICPC+1
023302  104401                  .WORD   .$.
023304                          SBR     1$              ;LOOP ON ERROR.
000036                          MICPC=MICPC+1
023304  100400                  .WORD   .$.
023306                  4$:     MOVE    # 0,BREG                 ;
000037                          MICPC=MICPC+1
023306  000400                  .WORD   .$.
023310                          MOVE    INP1 <CSR11>,SPAD <0>   ;
000040                          MICPC=MICPC+1
023310  123220                  .WORD   .$.
023312                          AND     SPAD <0>,BREG,OUT1 <CSR11>      ;RESET NON-EX-MEM.
000041                          MICPC=MICPC+1
023312  061271                  .WORD   .$.
023314                          CALL    SCPE            ;SCOPE THE TEST.
023314                          MOVE    # <MICPC+3>,BREG
000042                          MICPC=MICPC+1
023314  000444                  .WORD   .$.
023316                          SBR     SCPE
000043                          MICPC=MICPC+1
023316  104454                  .WORD   .$.
023320                          SBR     1$              ;DO NEXT ITERATION.
000044                          MICPC=MICPC+1
023320  100400                  .WORD   .$.
023322                  $NPR4
023322                  $XZ


                                ;*************************** TEST 35 ****************************
                                ;*NPR TEST
                                ;*TEST OF MULTIPLE NPR'S DOING DATI.
                                ;*XFER 6 WORD'S FROM 11 MEMORY TO UPROC.
                                ;*
023322                          $XZ
                                ;;*************************************************************************
023322                          $TSTN
                                ;  TEST 35
                                ;-------------
023322  012737  000035  001202  TST35:  MOV     #35,$TSTNM              ; LOAD THE NO. OF THIS TEST
023330  012737  023606  001442          MOV     #TST36,NEXT             ; POINT TO THE START OF NEXT TEST.
                                ;R1 CONTAINS BASE KMC11 ADDRESS
023336  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
023342  023356                  MCT35
023344  104022                  ERROR   22              ;TIME OUT ERROR...
023346  012706  001200          MOV     #STACK,SP       ;RESET STACK...
023352  000177  156064          JMP     @NEXT           ;GO TO NEXT TEST...
023356                  MCT35:
023356                  1$:     MOVE    # 5,SPAD <5>     ;SET THE COUNT
000000                          MICPC=MICPC+1
023356  003005                  .WORD   .$.
```

```
7900 023360
7901          000001
7902 023360   003013
7903 023362
7904          000002
7905 023362   000400
7906 023364
7907          000003
7908 023364   063236
7909 023366
7910          000004
7911 023366   062220
7912 023370
7913          000005
7914 023370   062221
7915 023372
7916          000006
7917 023372   062224
7918 023374
7919          000007
7920 023374   062225
7921 023376
7922          000010
7923 023376   061230
7924 023400
7925          000011
7926 023400   000572
7927 023402
7930          000012
7929 023402   063221
7930 023404
7931          000013
7932 023404   000447
7933 023406
7934          000014
7935 023406   063222
7936 023410
7937          000015
7938 023410   000525
7939 023412
7940          000016
7941 023412   010000
7942 023414
7943          000017
7944 023414   004002
7945 023416
7946          000020
7947 023416   076620
7948 023420
7949          000021
7950 023420   061620
7951 023422
7952          000022
7953 023422   063173
7954 023424
7955          000023
```

```
                MOVE    # 13,SPAD <13>  ;
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 0,BREG        ;SET TO CLEAR NPR IBNS+REGISTERS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUTO <0>   ;CLEAR IBUS
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUTO <1>   ;CLEAR IBUS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUTO <4>   ;CLEAR IBUS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUTO <5>   ;CLEAR IBUS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 <CSR10>       ;CLEAR EA+BITS
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # <MNPDT&377>,BREG      ;SET NPR ADDRESS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD <1>   ;
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # <MNPDT&177400/400>,BREG      ;
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,SPAD <2>           ;
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 125,BREG             ;GET READY TO LOAD MEMORY.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 0,MLR                ;
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 2,MPR        ;
                MICPC=MICPC+1
                .WORD   .$.
125:            MOVE    BREG,MEM,MARINC         ;LOAD THE MEMORY...
                MICPC=MICPC+1
                .WORD   .$.
                SHFBRT                          ;SET ALTERNATE 125 OR 252.
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                SDEC    SPAD <13>               ;IS IT DONE???
                MICPC=MICPC+1
.WORD           .$.!.DSP
                BZ      15$                     ;YES, START THE TEST.
                MICPC=MICPC+1
```

```
7956  023424  101425                .WORD   .$.
7957  023426                        SBR     12$                     ;NO, CONTINUE.
                                    MICPC=MICPC+1
7958          000024                .WORD   .$.
7959  023426  100420        15$:    MOVE    # 0,MLR         ;
7960  023430                        MICPC=MICPC+1
7961          000025                .WORD   .$.
7962  023430  010000                MOVE    # 0,MPR         ;
7963  023432                        MICPC=MICPC+1
7964          000026                .WORD   .$.
7965  023432  004000        2$:     MOVE    SPAD <1>,BREG   ;SET IN BA ADDRESS.
7966  023434                        MICPC=MICPC+1
7967          000027                .WORD   .$.
7968  023434  060601                MOVE    BREG,OUTO <4>   ;SET IN+BA ADDRESS.
7969  023436                        MICPC=MICPC+1
7970          000030                .WORD   .$.
7971  023436  062224                MOVE    SPAD <2>,BREG   ;SET IN+BA ADDRESS.
7972  023440                        MICPC=MICPC+1
7973          000031                .WORD   .$.
7974  023440  060602                MOVE    BREG,OUTO <5>   ;SET IN+BA ADDRESS.
7975  023442                        MICPC=MICPC+1
7976          000032                .WORD   .$.
7977  023442  062225                MOVE    # 03,BREG       ;SET NOT+LAST+XFR & NPR.R1
7978  023444                        MICPC=MICPC+1
7979          000033                .WORD   .$.
7980  023444  000403                MOVE    BREG,OUT1 <CSR10>       ;SET NOT+LAST+XFR & NPR R1 & WRB.XFR.
7981  023446                        MICPC=MICPC+1
7982          000034                .WORD   .$.
7983  023446  061230        3$:     MOVE    INP1 <CSR10>,BREG       ;IS NPR DONE??
7984  023450                        MICPC=MICPC+1
7985          000035                .WORD   .$.
7986  023450  120600                BBO     3$                      ;NO,SPIN ON IT
7987  023452                        MICPC=MICPC+1
7988          000036                .WORD   .$.
7989  023452  102035                MOVE    INP0 <0>,MEM MARINC     ;
7990  023454                        MICPC=MICPC+1
7991          000037                .WORD   .$.
7992  023454  036400                MOVE    INP0 <1>,MEM MARINC     ;
7993  023456                        MICPC=MICPC+1
7994          000040                .WORD   .$.
7995  023456  036420                SDEC    SPAD <5>        ;IS IT DONE??
7996  023460                        MICPC=MICPC+1
7997          000041        .WORD   .$.!.DSP
7998  023460  063165                BZ      4$                      ;YES, GO CHECK THE DATA.
7999  023462                        MICPC=MICPC+1
8000          000042                .WORD   .$.
8001  023462  101447                SINC    SPAD <1>                ;UPDATE ADDRESS.
8002  023464                        MICPC=MICPC+1
8003          000043        .WORD   .$.!.DSP
8004  023464  063061                SINC    SPAD <1>                ;
8005  023466                        MICPC=MICPC+1
8006          000044        .WORD   .$.!.DSP
8007  023466  063061                SADC    SPAD <2>                ;UPDATE ADDRESS.
8008  023470                        MICPC=MICPC+1
8009          000045        .WORD   .$.!.DSP
8010  023470  063102                SBR     2$                      ;DO NEXT NPR.
8011  023472
```

```
8012                000046
8013  023472        100427                    MICPC=MICPC+1
8014  023474                                  .WORD   .$.
8015                000047          45:       MOVE    # 0,BREG                ;MOVE # 5,SPAD <8>
8016  023474        000400                    MICPC=MICPC+1
8017  023476                                  .WORD   .$.
8018                000050                    MOVE    # 13,SPAD <13>          ;
8019  023476        003013                    MICPC=MICPC+1
8020  023500                                  .WORD   .$.
8021                000051                    MOVE    BREG,OUT1 <CSR10>       ;CLEAR NOT-LAST-XFR & NPR RQ.
8022  023500        061230                    MICPC=MICPC+1
8023  023502                                  .WORD   .$.
8024                000052                    MOVE    BREG,SPAD <3>   ;
8025  023502        063223                    MICPC=MICPC+1
8026  023504                                  .WORD   .$.
8027                000053                    MOVE    BREG,SPAD <0>
8028  023504        063220                    MICPC=MICPC+1
8029  023506                                  .WORD   .$.
8030                000054                    MOVE    BREG,SPAD <1>
8031  023506        063221                    MICPC=MICPC+1
8032  023510                                  .WORD   .$.
8033                000055                    MOVE    # 2,SPAD <2>    ;
8034  023510        003002                    MICPC=MICPC+1
8035  023512                                  .WORD   .$.
8036                000056                    MOVE    # 0,MLR         ;RESET MAR.
8037  023512        010000                    MICPC=MICPC+1
8038  023514                                  .WORD   .$.
8039                000057                    MOVE    # 0,MPR         ;RESTE MAR.
8040  023514        004000                    MICPC=MICPC+1
8041  023516                                  .WORD   .$.
8042                000060          55:       MOVE    MEM,SPAD <0>    ;GET THE NPR DATA.
8043  023516        043220                    MICPC=MICPC+1
8044  023520                                  .WORD   .$.
8045                000061                    MOVE    SPAD <1>,MLR    ;SET GOOD DATA ADDRESS.
8046  023520        070201                    MICPC=MICPC+1
8047  023522                                  .WORD   .$.
8048                000062                    MOVE    SPAD <2>,MPR    ;SET GOOD DATA ADDRESS.
8049  023522        064202                    MICPC=MICPC+1
8050  023524                                  .WORD   .$.
8051                                        SIFEQ   MEM,SPAD <0> 6$ ;IF GOOD CHECK NEXT...
8052
8053  023524                                SUB2C   SPAD <0>,MEM,NOP
8054                000063                  MICPC=MICPC+1
8055  023524        040360                  .WORD   .$.
8056  023526                                BZ      6$
8057                000064                  MICPC=MICPC+1
8058  023526        101475                  .WORD   .$.
8059  023530                                MOVE    MEM,OUT1 <CSR4>         ;GOOD DATA.
8060                000065                  MICPC=MICPC+1
8061  023530        041224                  .WORD   .$.
8062  023532                                MOVE    SPAD <0>,BREG   ;BAD DATA.
8063                000066                  MICPC=MICPC+1
8064  023532        060600                  .WORD   .$.
8065  023534                                MOVE    BREG,OUT1 <CSR5>        ;
8066                000067                  MICPC=MICPC+1
8067  023534        061225                  .WORD   .$.
```

```
023536                          MOVE    # 11,MEM        ;ERROR TYPE...
000070                          MICPC=MICPC+1
023536 002411                   .WORD   .$.
023540                          MOVE    MEM,OUT1 <CSR3>;
000071                          MICPC=MICPC+1
023540 041223                   .WORD   .$.
023542                          CALL    EROR1           ;DATA ERROR...
023542                          MOVE    # <MICPC+3>,BREG
000072                          MICPC=MICPC+1
023542 000474                   .WORD   .$.
023544                          SBR     ERROR1
000073                          MICPC=MICPC+1
023544 104401                   .WORD   .$.
023546                          SBR     1$              ;LOOP ON ERROR...
000074                          MICPC=MICPC+1
023546 100400                   .WORD   .$.
023550                  6$:     $DEC    SPAD <13>               ;IS IT DONE??
000075                          MICPC=MICPC+1
023550 063173           .WORD   .$.!.DSP
023552                          BZ      7$              ;YES, SCOPE IT.
000076                          MICPC=MICPC+1
023552 101503                   .WORD   .$.
023554                          $INC    SPAD <3>,,MLRLD ;NPR DATA ADDRESS.
000077                          MICPC=MICPC+1
023554 073063           .WORD   .$.!.DSP!MLRLD
023556                          $INC    SPAD <1>                ;UPDATE GOOD DATA ADDRESS.
000100                          MICPC=MICPC+1
023556 063061           .WORD   .$.!.DSP
023560                          $ADC    SPAD <2>                ;UPDATA GOOD DATA ADDRESS.
000101                          MICPC=MICPC+1
023560 063102           .WORD   .$.!.DSP
023562                          $BR     5$              ;CHECK NEXT BYTE.
000102                          MICPC=MICPC+1
023562 100460                   .WORD   .$.
023564                  7$:     CALL    SCPE            ;SCOPE IT.
023564                          MOVE    # <MICPC+3>,BREG
000103                          MICPC=MICPC+1
023564 000505                   .WORD   .$.
023566                          SBR     SCPE
000104                          MICPC=MICPC+1
023566 104454                   .WORD   .$.
023570                          SBR     1$              ;DO NEXT ITERATION.
000105                          MICPC=MICPC+1
023570 100400                   .WORD   .$.
023572 125125 125125 125125 MNPDT:  .WORD   125125,125125,125125        ;
023600 125125 125125 125125         .WORD   125125,125125,125125        ;
                        ;MNPDT1:            .BLKW   5
023606                  $MEM10
023606                  $XZ


                        ;************************** TEST 36 **************************
                        ;* MAIN MEMORY TEST
                        ;* FLOAT A 0 THROUGH ALL MAIN MEMORY LOCATION...
023606                  $XZ
                        ;;**********************************************************************
```

```
8124
8125  023606                                  $TSTN
8126                                                  ;  TEST 36
8127                                                  ;--------------
8128  023606  012737  000036  001202  TST36:  MOV     #36,$TSTNM          ; LOAD THE NO. OF THIS TEST
8129  023614  012737  023764  001442           MOV     #TST37,NEXT         ; POINT TO THE START OF NEXT TEST.
8130                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
8131  023622  004737  035536                    JSR     PC,LDVRWT           ;LOAD-VERIFY-WAIT.
8132  023626  023642                           MCT36
8133  023630  104022                           ERROR   22                  ;TIME OUT ERROR...
8134  023632  012706  001200                    MOV     #STACK,SP           ;RESET STACK...
8135  023636  000177  155600                    JMP     @NEXT               ;GO TO NEXT TEST...
8136  023642                           MCT36:
8137  023642                           $MAFLT  0,11$,12$,13$,14$,15$
8138  023642                                   MOVE    # 0,BREG            ;SET TO CLEAR SPAD 16...
8139                    000000
8140  023642  000400                           MICPC=MICPC+1
8141  023644                                   .WORD   .$.
8142                    000001                 MOVE    BREG,SPAD (16)      ;FOR RETURN ADDRESS PURPOSES...
8143  023644  063236                           MICPC=MICPC+1
8144  023646                                   .WORD   .$.
8145                    000002         11$:    MOVE    # 177,BREG          ;START WITH BIT 7.
8146  023646  000577                           MICPC=MICPC+1
8147  023650                                   .WORD   .$.
8148                    000003                 MOVE    # 0,MLR                     ;LOAD MAR<-LO.
8149  023650  010000                           MICPC=MICPC+1
8150  023652                                   .WORD   .$.
8151                    000004                 MOVE    # 0,MPR                     ;LOAD MAR<-HI.
8152  023652  004000                           MICPC=MICPC+1
8153  023654                                   .WORD   .$.
8154                                           MOVE    # 0,MEM
8155                    000005                 MICPC=MICPC+1
8156  023654  002400                           .WORD   .$.
8157  023656                                   MOVE    MEM,SPAD (1)        ;SET THE COUNTER...
8158                    000006                 MICPC=MICPC+1
8159  023656  043221                           .WORD   .$.
8160  023660                                   MOVE    MEM,SPAD (3)        ;SET THE COUNTER...
8161                    000007                 MICPC=MICPC+1
8162  023660  043223                           .WORD   .$.
8163
8164  023662                           12$:    MOVE    BREG,MEM            ;SET THE BIT IN MEMORY.
8165                    000010                 MICPC=MICPC+1
8166  023662  062620                           .WORD   .$.
8167  023664                                   MOVE    MEM,SPAD (2)        ;PUT THE "FOUND" IN SPAD (2).
8168                    000011                 MICPC=MICPC+1
8169  023664  043222                           .WORD   .$.
8170  023666                                   MOVE    BREG,SPAD (4)       ;SAVE THE BREG.
8171                    000012                 MICPC=MICPC+1
8172  023666  063224                           .WORD   .$.
8173  023670                                   $IFEQ   BREG,SPAD (2)   13$ ;IF GOOD CONTINUE.
8174
8175  023670
8176                    000013         SUB2C   SPAD (2),BREG,NOP
8177  023670  060362                           MICPC=MICPC+1
8178  023672                                   .WORD   .$.
8179                    000014                 BZ      13$
                                               MICPC=MICPC+1
```

```
8180  023672  101427
8181  023674
8182          000015
8183  023674  061224
8184  023676
8185          000016
8186  023676  041225
8187  023700
8188          000017
8189  023700  000406
8190  023702
8191          000020
8192  023702  061223
8193  023704
8194          000021
8195  023704  060601
8196  023706
8197          000022
8198  023706  061227
8199  023710
8200  023710
8201          000023
8202  023710  000425
8203  023712
8204          000024
8205  023712  104400
8206  023714
8207          000025
8208  023714  060604
8209  023716
8210          000026
8211  023716  100410
8212  023720
8213  023720
8214          000027
8215  023720  000431
8216  023722
8217          000030
8218  023722  104427
8219  023724
8220          000031
8221  023724  060604
8222  023726
8223          000032
8224  023726  100410
8225  023730
8226          000033
8227  023730  060604
8228  023732
8229          000034
8230  023732  061620
8231  023734
8232          000035
8233  023734  103410
8234  023736
8235          000036

                .WORD   .$.
                MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,OUT1 <CSR5> ;BAD DATA...
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    # 6,BREG                ;SET TYPE OF ERROR...
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 <CSR3>        ;
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <1>,BREG           ;LOAD ADDRESS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 <CSR7>        ;
                MICPC=MICPC+1
                .WORD   .$.
                CALL    ERROR                   ;REPORT ERROR...
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     ERROR
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG           ;RESTORE BREGISTER...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     12$                     ;LOOP ON ERROR...
                MICPC=MICPC+1
                .WORD   .$.
135:            CALL    SCP1                    ;LOOP ON DATA SET??
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG           ;YES, FIRST RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     12$                     ;SCOPE THE DATA..
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SHFRT                           ;RIGHT SHIFT BY 1 BIT.
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
887             12$             ;BRANCH IF NOT DONE.
                MICPC=MICPC+1
                .WORD   .$.
145:            $INC    SPAD <1>                ;INCREMENT ADDRESS.
                MICPC=MICPC+1
```

```
                        .WORD   .$.!.DSP
                        $AOC    SPAD (3)                        ;  "     "     .
                        MICPC=MICPC+1
                        .WORD   .$.!.DSP
                        MOVE    # 4,MEM                 ;PREPARE TO CHECK DONE...
                        MICPC=MICPC+1
                        .WORD   .$.
                        $IFLO   MEM,SPAD (3)    15$     ;IS IT DONE???

                        SUB2C   SPAD (3),MEM,NOP
                        MICPC=MICPC+1
                        .WORD   .$.
                        BC      15$
                        MICPC=MICPC+1
                        .WORD   .$.

                        MOVE    SPAD (1),MLR            ;LOAD MAR<-LO.
                        MICPC=MICPC+1
                        .WORD   .$.
                        MOVE    SPAD (3),MPR            ;LOAD MAR<-HI.
                        MICPC=MICPC+1
                        .WORD   .$.
                        $BR     12$                     ;DO THE NEXT MEMORY LOCATION.
                        MICPC=MICPC+1
                        .WORD   .$.
15$:                    CALL    SCPE                    ;SCOPE THE TEST...
                        MOVE    # (MICPC+3),BREG
                        MICPC=MICPC+1
                        .WORD   .$.
                        $BR     SCPE
                        MICPC=MICPC+1
                        .WORD   .$.
                        $BR     11$
                        MICPC=MICPC+1
                        .WORD   .$.
SMEM11
$XZ


                        ;*************************** TEST 37 ****************************
                        ;* MAIN MEMORY TEST
                        ;* FLOAT A 1 THROUGH ALL MAIN MEMORY LOCATIONS...
$XZ                     ;:***************************************************************
$TSTN
                        ;  TEST 37
                        ;-------------
023764 012737 000037 001202 TST37:  MOV     #37,$TSTNM              ; LOAD THE NO. OF THIS TEST
023772 012737 024144 001442         MOV     #TST40,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                            ;R1 CONTAINS BASE KMC11 ADDRESS
024000 004737 035536                JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
024004 024020                       MCT37
024006 104022                       ERROR   22                      ;TIME OUT ERROR...
024010 012706 001200                MOV     #STACK,SP               ;RESET STACK...
```

```
8292  024014  000177  155422          JMP     @NEXT                   ;GO TO NEXT TEST...
8293  024020                  MCT37:
8294  024020                  SWAFLT  1,1$,2$,3$,4$,5$
8295  024020                          MOVE    # 0,BREG        ;SET TO CLEAR SPAD 16...
8296          000000                  MICPC=MICPC+1
8297  024020  000400                  .WORD   .$.
8298  024022                          MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
8299          000001                  MICPC=MICPC+1
8300  024022  063236                  .WORD   .$.
8301  024024                  1$: MOVE        # 200,BREG       ;START WITH BIT 7.
8302          000002                  MICPC=MICPC+1
8303  024024  000600                  .WORD   .$.
8304  024026                          MOVE    # 0,MLR                 ;LOAD MAR<-LO.
8305          000003                  MICPC=MICPC+1
8306  024026  010000                  .WORD   .$.
8307  024030                          MOVE    # 0,MPR                 ;LOAD MAR<-HI.
8308          000004                  MICPC=MICPC+1
8309  024030  004000                  .WORD   .$.
8310  024032                          MOVE    # 0,MEM
8311          000005                  MICPC=MICPC+1
8312  024032  002400                  .WORD   .$.
8313  024034                          MOVE    MEM,SPAD <1>            ;SET THE COUNTER...
8314          000006                  MICPC=MICPC+1
8315  024034  043221                  .WORD   .$.
8316  024036                          MOVE    MEM,SPAD <3>            ;SET THE COUNTER...
8317          000007                  MICPC=MICPC+1
8318  024036  043223                  .WORD   .$.
8319
8320  024040                  2$:     MOVE    BREG,MEM                ;SET THE BIT IN MEMORY.
8321          000010                  MICPC=MICPC+1
8322  024040  062620                  .WORD   .$.
8323  024042                          MOVE    MEM,SPAD <2>            ;PUT THE "FOUND" IN SPAD  <2>.
8324          000011                  MICPC=MICPC+1
8325  024042  043222                  .WORD   .$.
8326  024044                          MOVE    BREG,SPAD <4>           ;SAVE THE BREG.
8327          000012                  MICPC=MICPC+1
8328  024044  063224                  .WORD   .$.
8329  024046                          $IFEQ   BREG,SPAD <2>   3$      ;IF GOOD CONTINUE.
8330
8331
8332  024046                          SUB2C   SPAD <2>,BREG,NOP
8333          000013                  MICPC=MICPC+1
8334  024046  060362                  .WORD   .$.
8335  024050                          BZ      3$
8336          000014                  MICPC=MICPC+1
8337  024050  101427                  .WORD   .$.
8338  024052                          MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA...
8339          000015                  MICPC=MICPC+1
8340  024052  061224                  .WORD   .$.
8341  024054                          MOVE    MEM,OUT1 <CSR5> ;BAD DATA...
8342          000016                  MICPC=MICPC+1
8343  024054  041225                  .WORD   .$.
8344  024056                          MOVE    # 6,BREG                ;SET TYPE OF ERROR...
8345          000017                  MICPC=MICPC+1
8346  024056  000406                  .WORD   .$.
8347  024060                          MOVE    BREG,OUT1 <CSR3>        ;
```

```
8348                000020
8349  024060  061223
8350  024062
8351                000021
8352  024062  060601
8353  024064
8354                000022
8355  024064  061227
8356  024066
8357  024066
8358                000023
8359  024066  000425
8360  024070
8361                000024
8362  024070  104400
8363  024072
8364                000025
8365  024072  060604
8366  024074
8367                000026
8368  024074  100410
8369  024076
8370  024076
8371                000027
8372  024076  000431
8373  024100
8374                000030
8375  024100  104427
8376  024102
8377                000031
8378  024102  060604
8379  024104
8380                000032
8381  024104  100410
8382  024106
8383                000033
8384  024106  060604
8385  024110
8386                000034
8387  024110  061620
8388  024112
8389                000035
8390  024112  103437
8391  024114
8392                000036
8393  024114  100410
8394  024116
8395                000037
8396  024116  063061
8397  024120
8398                000040
8399  024120  063103
8400  024122
8401                000041
8402  024122  002404
8403  024124
```

```
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <1>,BREG           ;LOAD ADDRESS.
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    BREG,OUT1 <CSR7>        ;
                MICPC=MICPC+1
                .WORD   .$.
                CALL    ERROR                   ;REPORT ERROR...
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     ERROR
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG           ;RESTORE BREGISTER...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;LOOP ON ERROR...
                MICPC=MICPC+1
                .WORD   .$.
3$:             CALL    SCP1                    ;LOOP ON DATA SET??
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCP1
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG           ;YES, FIRST RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;SCOPE THE DATA..
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    SPAD <4>,BREG           ;RESTORE BREG...
                MICPC=MICPC+1
                .WORD   .$.
                SHFRT                           ;RIGHT SHIFT BY 1 BIT.
                MICPC=MICPC+1
                .WORD   .SBR!.SELB!.DBRSH
                BB7     4$                      ;BRANCH IF DONE...
                MICPC=MICPC+1
                .WORD   .$.
                SBR     2$                      ;ELSE, CONTINUE.
                MICPC=MICPC+1
                .WORD   .$.
4$:             SINC    SPAD <1>                        ;INCREMENT ADDRESS.
                MICPC=MICPC+1
.WORD           .$.!.DSP
                SADC    SPAD <3>                        ;  "     "       .
                MICPC=MICPC+1
.WORD           .$.!.DSP
                MOVE    # 4,MEM                 ;PREPARE TO CHECK DONE...
                MICPC=MICPC+1
                .WORD   .$.
                SIFLO   MEM,SPAD <3>    5$      ;IS IT DONE???
```

```
                                        SUB2C   SPAD (3),MEM,NOP
024124                                  MICPC=MICPC+1
        000042
024124  040363                          .WORD   .$.
024126                                  BC      5$
        000043                          MICPC=MICPC+1
024126  101047                          .WORD   .$.

024130                                  MOVE    SPAD (1),MLR            ;LOAD MAR+LO.
        000044                          MICPC=MICPC+1
024130  070201                          .WORD   .$.
024132                                  MOVE    SPAD (3),MPR            ;LOAD MAR+HI.
        000045                          MICPC=MICPC+1
024132  064203                          .WORD   .$.
024134                                  SBR     2$                      ;DO THE NEXT MEMORY LOCATION.
        000046                          MICPC=MICPC+1
024134  100410                          .WORD   .$.
024136                          5$:     CALL    SCPE                    ;SCOPE THE TEST...
024136                                  MOVE    @ (MICPC+3),BREG
        000047                          MICPC=MICPC+1
024136  000451                          .WORD   .$.
024140                                  SBR     SCPE
        000050                          MICPC=MICPC+1
024140  104454                          .WORD   .$.
024142                                  SBR     1$
        000051                          MICPC=MICPC+1
024142  100402                          .WORD   .$.
024144                          SMEM2
024144                          SXZ


                                        ;************************** TEST 40 ****************************
                                        ;* MAIN MEMORY DUAL ADDRESSING TEST
                                        ;* LOAD EACH MEMORY LOCATION WITH IT'S OWN PAGE LESS ADDRESS.
                                        ;* READ BACK EACH LOCATION TO VERIFY CORRECT ADDRESSING.
024144                          SXZ     ;;***********************************************************************
024144                          STSTN
                                        ;  TEST 40
                                        ;-------------
024144  012737  000040  001202  TST40:  MOV     #40,$TSTNM              ; LOAD THE NO. OF THIS TEST
024152  012737  024362  001442          MOV     #TST41,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
024160  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
024164  024200                          MCT40
024166  104022                          ERROR   22                      ;TIME OUT ERROR...
024170  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
024174  000177  155242                  JMP     @NEXT                   ;GO TO NEXT TEST...
024200                          MCT40:
024200                          6$:     MOVE    # 0,BREG                ;START AT ADDRESS 0.
        000000                          MICPC=MICPC+1
024200  000400                          .WORD   .$.
024202                                  MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS PURPOSE...
        000001                          MICPC=MICPC+1
```

```
6460  024202  063236          .WORD   .S.
6461  024204                  MOVE    #0,MLR          ;LOAD MAR<-LO.
6462          000002          MICPC=MICPC+1
6463  024204  010000          .WORD   .S.
6464  024206                  MOVE    #0,MPR          ;LOAD MAR<-HI.
6465          000003          MICPC=MICPC+1
6466  024206  004000          .WORD   .S.
6467  024210                  MOVE    BREG,SPAD <1>   ;SET THE PARALLEL ADDRESS.
6468          000004          MICPC=MICPC+1
6469  024210  063221          .WORD   .S.
6470  024212                  MOVE    BREG,SPAD <2>   ;SET THE PARALLEL ADDRESS.
6471          000005          MICPC=MICPC+1
6472  024212  063222          .WORD   .S.
6473  024214          1$:     MOVE    SPAD <1>,MEM    ;SET THE ADDRESS IN MEMORY.
6474          000006          MICPC=MICPC+1
6475  024214  062601          .WORD   .S.
6476  024216                  MOVE    MEM,BREG        ;PUT FOUND IN BREG.
6477          000007          MICPC=MICPC+1
6478  024216  040620          .WORD   .S.
6479  024220                  SIFEQ   BREG,SPAD <1>   2$      ;BRANCH IF GOOD...
6480
6481
6482  024220                  SUB2C   SPAD <1>,BREG,NOP
6483          000010          MICPC=MICPC+1
6484  024220  060361          .WORD   .S.
6485  024222                  BZ      2$
6486          000011          MICPC=MICPC+1
6487  024222  101424          .WORD   .S.
6488  024224                  MOVE    MEM,OUT1 <CSR5>         ;LOAD GOOD DATA...
6489          000012          MICPC=MICPC+1
6490  024224  041225          .WORD   .S.
6491  024226                  MOVE    SPAD <1>,BREG           ;LOAD BAD DATA...
6492          000013          MICPC=MICPC+1
6493  024226  060601          .WORD   .S.
6494  024230                  MOVE    BREG,OUT1 <CSR4>        ;
6495          000014          MICPC=MICPC+1
6496  024230  061224          .WORD   .S.
6497  024232                  MOVE    #7,BREG                 ;SET THE ERROR TYPE...
6498          000015          MICPC=MICPC+1
6499  024232  000407          .WORD   .S.
6500  024234                  MOVE    BREG,OUT1 <CSR3>        ;
6501          000016          MICPC=MICPC+1
6502  024234  061223          .WORD   .S.
6503  024236                  MOVE    SPAD <1>,BREG           ;LOAD ADDRESS.
6504          000017          MICPC=MICPC+1
6505  024236  060601          .WORD   .S.
6506  024240                  MOVE    BREG,OUT1 <CSR7>        ;
6507          000020          MICPC=MICPC+1
6508  024240  061227          .WORD   .S.
6509  024242                  CALL    ERROR1                  ;REPORT ERROR...
6510  024242                  MOVE    @ <MICPC+3>,BREG
6511          000021          MICPC=MICPC+1
6512  024242  000423          .WORD   .S.
6513  024244                  SBR     ERROR1
6514          000022          MICPC=MICPC+1
6515  024244  104401          .WORD   .S.
```

```
                                        SBR     1$                              ;LOOP ON ERROR...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                2$:     CALL    SCP11                           ;SCOPE ON DATA SET?
                                        MOVE    @ <MICPC+3>,BREG
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SBR     SCP11
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SBR     1$                              ;YES, DO IT...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SINC    SPAD <1>,       ,MLRLD          ;INCREMENT THE ADDRESS.
                                        MICPC=MICPC+1
                                .WORD   .$.!.DSP!MLRLD
                                        SADC    SPAD <2>,       ,MPRLD          ;INCREMENT PAGE ADDRESS.
                                        MICPC=MICPC+1
                                .WORD   .$.!.DSP!MPRLD
                                        MOVE    @ 4,BREG                        ;CHECK IF DONE?
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SIFHI   BREG,SPAD <2>   1$              ;NO, THEN CONTINUE...


                                        SUB1C   SPAD <2>,BREG,NOP
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        BC      64$
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SBR     1$
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                64$:
                                3$:     MOVE    @ 0,BREG                        ;PREPARE TO CHECK IT.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    @ 0,MLR                 ;LOAD MAR+LO.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    @ 0,MPR                 ;LOAD MAR+HI.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    BREG,SPAD <1>           ;SET THE PARALLEL ADDRESS.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        MOVE    BREG,SPAD <2>           ;SET THE PARALLEL ADDRESS.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                4$:     MOVE    MEM,BREG                ;PUT "FOUND" IN BREG.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SIFEQ   BREG,SPAD <1>   5$      ;BRANCH IF GOOD...
```

```
024306                  SUB2C   SPAD (1),BREG,NOP
000043                  MICPC=MICPC+1
024306 060361           .WORD   .$.
024310                  BZ      5$
000044                  MICPC=MICPC+1
024310 101455           .WORD   .$.
024312                  MOVE    MEM,OUT1 (CSR5)         ;
000045                  MICPC=MICPC+1
024312 041225           .WORD   .$.
024314                  MOVE    SPAD (1),BREG           ;
000046                  MICPC=MICPC+1
024314 060601           .WORD   .$.
024316                  MOVE    BREG,OUT1 (CSR4)        ;GOOD DATA...
000047                  MICPC=MICPC+1
024316 061224           .WORD   .$.
024320                  MOVE    #7,BREG                 ;SET ERROR TYPE.
000050                  MICPC=MICPC+1
024320 000407           .WORD   .$.
024322                  MOVE    BREG,OUT1 (CSR3)        ;
000051                  MICPC=MICPC+1
024322 061223           .WORD   .$.
024324                  CALL    ERROR1                  ;REPORT ERROR.
024324                  MOVE    #(MICPC+3),BREG
000052                  MICPC=MICPC+1
024324 000454           .WORD   .$.
024326                  SBR     ERROR1
000053                  MICPC=MICPC+1
024326 104401           .WORD   .$.
024330                  SBR     4$                      ;LOOP ON ERROR.
000054                  MICPC=MICPC+1
024330 100442           .WORD   .$.
024332          5$:     CALL    SCP11                   ;SCOPE THE DATA??
024332                  MOVE    #(MICPC+3),BREG
000055                  MICPC=MICPC+1
024332 000457           .WORD   .$.
024334                  SBR     SCP11
000056                  MICPC=MICPC+1
024334 104430           .WORD   .$.
024336                  SBR     4$                      ;YES, SCOPE IT.
000057                  MICPC=MICPC+1
024336 100442           .WORD   .$.
024340                  SINC    SPAD (1), ,MLRLD        ;INCREMENT MEMORY ADDRESS.
000060                  MICPC=MICPC+1
024340 073061   .WORD    .$.!.DSP!MLRLD
024342                  SADC    SPAD (2), ,MPRLD        ;INCREMENT MEMORY ADDRESS.
000061                  MICPC=MICPC+1
024342 067102   .WORD    .$.!.DSP!MPRLD
024344                  MOVE    #4,BREG                         ;CHECK IF DONE??
000062                  MICPC=MICPC+1
024344 000404           .WORD   .$.
024346                  $IFHI   BREG,SPAD (2)   4$      ;NO, THEN CONTINUE...


024346                  SUB1C   SPAD (2),BREG,NOP
000063                  MICPC=MICPC+1
024346 060342           .WORD   .$.
```

```
                                        BC      65$
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SBR     4$
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                65$:
                                        CALL    SCPE                    ;SCOPE THE TEST...
                                        MOVE    @ <MICPC+3>,BREG
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SBR     SCPE
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SBR     6$                      ;DO THE NEXT ITERATION.
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                SMEM31
                                $XZ


                                ;*************************** TEST 41 ****************************
                                ;* MAR TEST.
                                ;* PERFORM DUAL ADDRESSING TEST.
                                ;* USING MAR AUTO-INC FEATURE.
                                $XZ
                                ;;************************************************************************
                                $TSTN
                                ;  TEST 41
                                ;-------------
024362 012737 000041 001202     TST41:  MOV     #41,$TSTNM              ; LOAD THE NO. OF THIS TEST
024370 012737 024544 001442             MOV     #TST42,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
024376 004737 035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
024402 024416                           MCT41
024404 104022                           ERROR   22                      ;TIME OUT ERROR...
024406 012706 001200                    MOV     #STACK,SP               ;RESET STACK...
024412 000177 155024                    JMP     @NEXT                   ;GO TO NEXT TEST...
024416                          MCT41:
024416                          1$:     MOVE    @ 0,BREG                ;START WITH ZERO.
                                        MICPC=MICPC+1
024416 000400                           .WORD   .$.
024420                                  MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
                                        MICPC=MICPC+1
024420 063236                           .WORD   .$.
024422                                  MOVE    @ 0,MLR                 ;LOAD MAR+LO.
                                        MICPC=MICPC+1
024422 010000                           .WORD   .$.
024424                                  MOVE    @ 0,MPR                 ;LOAD MAR+HI.
                                        MICPC=MICPC+1
024424 004000                           .WORD   .$.
024426                                  MOVE    BREG,SPAD <1>           ;SET PARALLEL ADDRESS.
                                        MICPC=MICPC+1
024426 063221                           .WORD   .$.
024430                                  MOVE    BREG,SPAD <2>           ;SET PARALLEL ADDRESS.
```

```
                    000005                  MICPC=MICPC+1
024430  063222                              .WORD   .S.
024432                                      MOVE    # 4,BREG
                    000006                  MICPC=MICPC+1
024432  000404                              .WORD   .S.
024434                          2$:         MOVE    SPAD <1>,MEM,MARINC     ;LOAD DATA IN MEMORY.
                    000007                  MICPC=MICPC+1
024434  076601                              .WORD   .S.
024436                                      SINC    SPAD <1>                ;UPDATE ADDRESS.
                    000010                  MICPC=MICPC+1
024436  063061                  .WORD       .S.!.DSP
024440                                      SADC    SPAD <2>                ;UPDATE ADDRESS.
                    000011                  MICPC=MICPC+1
024440  063102                  .WORD       .S.!.DSP
024442                                      SIFHI   BREG,SPAD <2>   2$      ;BRANCH IF NOT DONE...

024442                                      SUB1C   SPAD <2>,BREG,NOP
                    000012                  MICPC=MICPC+1
024442  060342                              .WORD   .S.
024444                                      BC      64$
                    000013                  MICPC=MICPC+1
024444  101015                              .WORD   .S.
024446                                      SBR     2$
                    000014                  MICPC=MICPC+1
024446  100407                              .WORD   .S.
024450                          64$:
024450                                      MOVE    # 0,BREG                ;START WITH 0 AGAIN.
                    000015                  MICPC=MICPC+1
024450  000400                              .WORD   .S.
024452                                      MOVE    # 0,MLR                 ;LOAD MAR+LO.
                    000016                  MICPC=MICPC+1
024452  010000                              .WORD   .S.
024454                                      MOVE    # 0,MPR                 ;LOAD MAR+HI.
                    000017                  MICPC=MICPC+1
024454  004000                              .WORD   .S.
024456                                      MOVE    BREG,SPAD <1>           ;SET PARALLEL ADDRESS.
                    000020                  MICPC=MICPC+1
024456  063221                              .WORD   .S.
024460                                      MOVE    BREG,SPAD <2>           ;SET PARALLEL ADDRESS.
                    000021                  MICPC=MICPC+1
024460  063222                              .WORD   .S.
024462                          3$:         MOVE    MEM,BREG,MARINC         ;GET THE CONTENTS OF MEMORY
                    000022                  MICPC=MICPC+1
024462  054620                              .WORD   .S.
024464                                      SIFEQ   BREG,SPAD <1>   4$      ;BRANCH IF GOOD.

024464                                      SUB2C   SPAD <1>,BREG,NOP
                    000023                  MICPC=MICPC+1
024464  060361                              .WORD   .S.
024466                                      BZ      4$
                    000024                  MICPC=MICPC+1
024466  101436                              .WORD   .S.
024470                                      MOVE    MEM,OUT1 <CSR5>         ;GOOD DATA.
                    000025                  MICPC=MICPC+1
```

```
8740  024470  041225                .WORD   .$.
8741  024472                        MOVE    SPAD <1>,BREG           ;BAD DATA.
8742          000026                MICPC=MICPC+1
8743  024472  060601                .WORD   .$.
8744  024474                        MOVE    BREG,OUT1 <CSR4>        ;
8745          000027                MICPC=MICPC+1
8746  024474  061224                .WORD   .$.
8747  024476                        MOVE    # 10,BREG               ;SET THE ERROR TYPE.
8748          000030                MICPC=MICPC+1
8749  024476  000410                .WORD   .$.
8750  024500                        MOVE    BREG,OUT1 <CSR3>        ;
8751          000031                MICPC=MICPC+1
8752  024500  061223                .WORD   .$.
8753  024502                        CALL    ERROR                   ;REPORT ERROR.
8754  024502                        MOVE    # <MICPC+3>,BREG
8755          000032                MICPC=MICPC+1
8756  024502  000434                .WORD   .$.
8757  024504                        SBR     ERROR
8758          000033                MICPC=MICPC+1
8759  024504  104400                .WORD   .$.
8760  024506                        MOVE    SPAD <1>,MLR    ;LOOP ON ERROR.
8761          000034                MICPC=MICPC+1
8762  024506  070201                .WORD   .$.
8763  024510                        SBR     3$                      ;LOOP ON ERROR.
8764          000035                MICPC=MICPC+1
8765  024510  100422                .WORD   .$.
8766  024512                4$:     CALL    SCP1                    ;SCOPE THE DATA...
8767  024512                        MOVE    # <MICPC+3>,BREG
8768          000036                MICPC=MICPC+1
8769  024512  000440                .WORD   .$.
8770  024514                        SBR     SCP1
8771          000037                MICPC=MICPC+1
8772  024514  104427                .WORD   .$.
8773  024516                        MOVE    SPAD <1>,MLR            ;RESTORE POINTER...
8774          000040                MICPC=MICPC+1
8775  024516  070201                .WORD   .$.
8776  024520                        SBR     3$                      ;LOOP ON DATA...
8777          000041                MICPC=MICPC+1
8778  024520  100422                .WORD   .$.
8779  024522                        SINC    SPAD <1>                        ;UPDATE ADDRESS.
8780          000042                MICPC=MICPC+1
8781  024522  063061          .WORD .$.!.DSP
8782  024524                        SADC    SPAD <2>                        ;UPDATE ADDRESS.
8783          000043                MICPC=MICPC+1
8784  024524  063102          .WORD .$.!.DSP
8785  024526                        MOVE    # 4,BREG                        ;CHECK IF DONE.
8786          000044                MICPC=MICPC+1
8787  024526  000404                .WORD   .$.
8788  024530                        SIFHI   BREG,SPAD <2>   3$      ;IF NOT THEN CONTINUE...
8789
8790
8791  024530                        SUB1C   SPAD <2>,BREG,NOP
8792          000045                MICPC=MICPC+1
8793  024530  060342                .WORD   .$.
8794  024532                        BC      65$
8795          000046                MICPC=MICPC+1
```

```
8796  024532  101050                        .WORD   .$.
8797  024534                                $BR     .$
8798          000047                        MICPC=MICPC+1
8799  024534  100422                        .WORD   .$.
8800  024536                        65$:
8801  024536                                CALL    SCPE                    ;SCOPE THE TEST.
8802  024536                                MOVE    @ <MICPC+3>,BREG
8803          000050                        MICPC=MICPC+1
8804  024536  000452                        .WORD   .$.
8805  024540                                $BR     SCPE
8806          000051                        MICPC=MICPC+1
8807  024540  104454                        .WORD   .$.
8808  024542                                $BR     1$                      ;DO THE NEXT ITERATION...
8809          000052                        MICPC=MICPC+1
8810  024542  100400                        .WORD   .$.
8811  024544                        SALUTO
8812  024544                        $XZ
8813
8814
8815                                        ;************************** TEST 42 **************************
8816                                        ;*ALU C BIT TEST
8817                                        ;*TEST THAT ADD OF 377 AND 1 WILL SET TAG CBIT.
8818                                        ;*THEN CHECK IF C BIT CLEARS
8819  024544                        $XZ
8820                                        ;:*********************************************************************
8821
8822  024544                        $TSTN
8823                                        ;  TEST 42
8824                                        ;--------------
8825  024544  012737  000042  001202  TST42: MOV     #42,$TSTNM              ; LOAD THE NO. OF THIS TEST
8826  024552  012737  024700  001442        MOV     #TST43,NEXT             ; POINT TO THE START OF NEXT TEST.
8827                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
8828  024560  004737  035536                JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
8829  024564  024600                        MCT42
8830  024566  104022                        ERROR   22                      ;TIME OUT ERROR...
8831  024570  012706  001200                MOV     #STACK,SP               ;RESET STACK...
8832  024574  000177  154642                JMP     @NEXT                   ;GO TO NEXT TEST...
8833  024600                        MCT42:
8834  024600                        1$:     MOVE    @ 0,BREG        ;SET TO CLEAR REQUIRED LOCATIONS.
8835          000000                        MICPC=MICPC+1
8836  024600  000400                        .WORD   .$.
8837  024602                                MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES.
8838          000001                        MICPC=MICPC+1
8839  024602  063236                        .WORD   .$.
8840  024604                                MOVE    @ 0,MLR         ;
8841          000002                        MICPC=MICPC+1
8842  024604  010000                        .WORD   .$.
8843  024606                                MOVE    @ 0,MPR         ;
8844          000003                        MICPC=MICPC+1
8845  024606  004000                        .WORD   .$.
8846  024610                                MOVE    BREG,SPAD <0>   ;CLEAR SPAD 0.
8847          000004                        MICPC=MICPC+1
8848  024610  063220                        .WORD   .$.
8849  024612                                MOVE    BREG,SPAD <1>   ;CLEAR SPAD 1.
8850          000005                        MICPC=MICPC+1
8851  024612  063221                        .WORD   .$.
```

```
024614                          MOVE    BREG,SPAD (2)   ;CLEAR SPAD 2.
000006                          MICPC=MICPC+1
024614 063222                   .WORD   .$.
024616                          MOVE    # 377,MEM       ;
000007                          MICPC=MICPC+1
024616 002777                   .WORD   .$.
024620                          MOVE    MEM,SPAD (1)    ;LOAD 377 IN SPAD 1.
000010                          MICPC=MICPC+1
024620 043221                   .WORD   .$.
024622                          $INC    SPAD (1)                ;ADD 1 TO IT.
000011                          MICPC=MICPC+1
024622 063061           .WORD   .$.!.DSP
024624                          $ADC    SPAD (2)                ;GET THE CARRY BIT..
000012                          MICPC=MICPC+1
024624 063102           .WORD   .$.!.DSP
024626                          MOVE    SPAD (2),BREG   ;IS CARRY SET??
000013                          MICPC=MICPC+1
024626 060602                   .WORD   .$.
024630                          BBO     2$              ;YES SCOPE THE TEST DATA.
000014                          MICPC=MICPC+1
024630 102022                   .WORD   .$.
024632                  6$:     ;MOVE                   ;GOOD DATA.
024632                          MOVE    # 21,MEM                ;TYPE OF ERROR.
000015                          MICPC=MICPC+1
024632 002421                   .WORD   .$.
024634                          MOVE    MEM,OUT1 (CSR3)         ;
000016                          MICPC=MICPC+1
024634 041223                   .WORD   .$.
024636                          CALL    EROR1           ;REPORT ERROR...
024636                          MOVE    # (MICPC+3),BREG
000017                          MICPC=MICPC+1
024636 000421                   .WORD   .$.
024640                          $BR     EROR1
000020                          MICPC=MICPC+1
024640 104401                   .WORD   .$.
024642                          $BR     1$              ;LOOP ON ERROR...
000021                          MICPC=MICPC+1
024642 100400                   .WORD   .$.
024644                  2$:     CALL    SCP11           ;SCOPE THE DATA??
024644                          MOVE    # (MICPC+3),BREG
000022                          MICPC=MICPC+1
024644 000424                   .WORD   .$.
024646                          $BR     SCP11
000023                          MICPC=MICPC+1
024646 104430                   .WORD   .$.
024650                          $BR     1$              ;YES, DO IT...
000024                          MICPC=MICPC+1
024650 100400                   .WORD   .$.
024652                  3$:     $DEC    SPAD (2)                ;RESET THE SPAD  (2).
000025                          MICPC=MICPC+1
024652 063162           .WORD   .$.!.DSP
024654                          MOVE    # 377,MEM       ;LOAD 377 IN MEMORY.
000026                          MICPC=MICPC+1
024654 002777                   .WORD   .$.
024656                          MOVE    MEM,SPAD (1)    ;LOAD IN SPAD.  (1).
000027                          MICPC=MICPC+1
```

```
024656 043221                          .WORD   .$.
024660                                 $INC    SPAD <1>                ;ADD 1 TO IT.
000030                                 MICPC=MICPC+1
024660 063061                  .WORD   .$.!.DSP
024662                                 $DEC    SPAD <1>                ;SUBTRACT 1 OUT OF IT.
000031                                 MICPC=MICPC+1
024662 063161                  .WORD   .$.!.DSP
024664                                 $ADC    SPAD <0>        ;GET THE CARRY BIT.
000032                                 MICPC=MICPC+1
024664 063100                  .WORD   .$.!.DSP
024666                         45:     MOVE    SPAD <0>,BREG   ;WAS IT RESET?
000033                                 MICPC=MICPC+1
024666 060600                          .WORD   .$.
024670                                 BBO     6$              ;NO, REPORT ERROR.
000034                                 MICPC=MICPC+1
024670 102015                          .WORD   .$.
024672                                 CALL    SCPE            ;SCOPE THE TEST.
024672                                 MOVE    # <MICPC+3>,BREG
000035                                 MICPC=MICPC+1
024672 000437                          .WORD   .$.
024674                                 $BR     SCPE
000036                                 MICPC=MICPC+1
024674 104454                          .WORD   .$.
024676                                 $BR     1$              ;DO THE NEXT ITERATION.
000037                                 MICPC=MICPC+1
024676 100400                          .WORD   .$.
024700                         $ALUT1  0,<SEL B & SEL A>,MOVE,0,0,-1,-1,0,0,-1,-1,125,125,252,252,125,125,252,252,<B>,0
024700                         $XZ


                                       ;************************** TEST 43 ****************************
                                       ;#ALU TEST
                                       ;#TEST OF ALU FUNCTION SEL B & SEL A WITH C BIT CLEARED.
                                       ;#TEST OF ALU FUNCTION SEL B & SEL A WITH C BIT SET.
                                       ;#ALU FUNCTION (B)
                                       ;#LOAD MAIN MEMORY 16 WORDS OF DATA.
                                       ;#PERFORM THE FUNCTION, VERIFY THE RESULTS..
024700                         $XZ
                                       ;;*************************************************************************
024700                         $TSTN
                                       ;  TEST 43
                                       ;-------------
024700 012737 000043 001202    TST43:  MOV     #43,$TSTNM              ; LOAD THE NO. OF THIS TEST
024706 012737 025226 001442            MOV     #TST44,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                               ;R1 CONTAINS BASE KMC11 ADDRESS
024714 004737 035536                   JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
024720 024734                          MCT43
024722 104022                          ERROR   22                      ;TIME OUT ERROR...
024724 012706 001200                   MOV     #STACK,SP               ;RESET STACK...
024730 000177 154506                   JMP     @NEXT                   ;GO TO NEXT TEST...
024734                         MCT43:
024734                         1$:     MOVE    # 0,MLR          ;SET MAR<-LO.
000000                                 MICPC=MICPC+1
024734 010000                          .WORD   .$.
024736                                 MOVE    # 0,MPR          ;SET MAR<-HI.
```

```
8964                000001                  MICPC=MICPC+1
8965  024736  004000                        .WORD   .$.
8966  024740                                MOVE    # 0,BREG                ;
8967                000002                  MICPC=MICPC+1
8968  024740  000400                        .WORD   .$.
8969  024742                                MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
8970                000003                  MICPC=MICPC+1
8971  024742  063236                        .WORD   .$.
8972  024744                                MOVE    BREG,SPAD <0>   ;
8973                000004                  MICPC=MICPC+1
8974  024744  063220                        .WORD   .$.
8975  024746                                MOVE    BREG,SPAD <1>   ;
8976                000005                  MICPC=MICPC+1
8977  024746  063221                        .WORD   .$.
8978  024750                                MOVE    BREG,SPAD <2>   ;
8979                000006                  MICPC=MICPC+1
8980  024750  063222                        .WORD   .$.
8981  024752                                $DEC    SPAD <2>                ;
8982                000007                  MICPC=MICPC+1
6783  024752  063162            .WORD       .$.!.DSP
8984  024754                                MOVE    BREG,SPAD <4>           ;
8985                000010                  MICPC=MICPC+1
8986  024754  063224                        .WORD   .$.
8987  024756                                MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
8988                000011                  MICPC=MICPC+1
8989  024756  016400                        .WORD   .$.
8990  024760                                MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
8991                000012                  MICPC=MICPC+1
8992  024760  016400                        .WORD   .$.
8993  024762                                MOVE    # 0,MEM MARINC
8994                000013                  MICPC=MICPC+1
8995  024762  016400                        .WORD   .$.
8996  024764                                MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
8997                000014                  MICPC=MICPC+1
8998  024764  016400                        .WORD   .$.
8999  024766                                MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
9000                000015                  MICPC=MICPC+1
9001  024766  016777                        .WORD   .$.
9002  024770                                MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
9003                000016                  MICPC=MICPC+1
9004  024770  016400                        .WORD   .$.
9005  024772                                MOVE    # -1,MEM MARINC
9006                000017                  MICPC=MICPC+1
9007  024772  016777                        .WORD   .$.
9008  024774                                MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
9009                000020                  MICPC=MICPC+1
9010  024774  016777                        .WORD   .$.
9011  024776                                MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
9012                000021                  MICPC=MICPC+1
9013  024776  016400                        .WORD   .$.
9014  025000                                MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
9015                000022                  MICPC=MICPC+1
9016  025000  016777                        .WORD   .$.
9017  025002                                MOVE    # 0,MEM MARINC
9018                000023                  MICPC=MICPC+1
9019  025002  016400                        .WORD   .$.
```

```
9020 025004                     MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
9021         000024             MICPC=MICPC+1
9022 025004  016400             .WORD   .$.
9023 025006                     MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
9024         000025             MICPC=MICPC+1
9025 025006  016777             .WORD   .$.
9026 025010                     MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
9027         000026             MICPC=MICPC+1
9028 025010  016777             .WORD   .$.
9029 025012                     MOVE    # -1,MEM MARINC
9030         000027             MICPC=MICPC+1
9031 025012  016777             .WORD   .$.
9032 025014                     MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
9033         000030             MICPC=MICPC+1
9034 025014  016777             .WORD   .$.
9035 025016                     MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9036         000031             MICPC=MICPC+1
9037 025016  016525             .WORD   .$.
9038 025020                     MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9039         000032             MICPC=MICPC+1
9040 025020  016525             .WORD   .$.
9041 025022                     MOVE    # 125,MEM MARINC
9042         000033             MICPC=MICPC+1
9043 025022  016525             .WORD   .$.
9044 025024                     MOVE    # 125,MEM MARINC         ;RESULT WITH C BIT SET.
9045         000034             MICPC=MICPC+1
9046 025024  016525             .WORD   .$.
9047 025026                     MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9048         000035             MICPC=MICPC+1
9049 025026  016652             .WORD   .$.
9050 025030                     MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9051         000036             MICPC=MICPC+1
9052 025030  016525             .WORD   .$.
9053 025032                     MOVE    # 252,MEM MARINC
9054         000037             MICPC=MICPC+1
9055 025032  016652             .WORD   .$.
9056 025034                     MOVE    # 252,MEM MARINC         ;RESULT WITH C BIT SET.
9057         000040             MICPC=MICPC+1
9058 025034  016652             .WORD   .$.
9059 025036                     MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9060         000041             MICPC=MICPC+1
9061 025036  016525             .WORD   .$.
9062 025040                     MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY
9063         000042             MICPC=MICPC+1
9064 025040  016652             .WORD   .$.
9065 025042                     MOVE    # 125,MEM MARINC
9066         000043             MICPC=MICPC+1
9067 025042  016525             .WORD   .$.
9068 025044                     MOVE    # 125,MEM MARINC         ;RESULT WITH C BIT SET.
9069         000044             MICPC=MICPC+1
9070 025044  016525             .WORD   .$.
9071 025046                     MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9072         000045             MICPC=MICPC+1
9073 025046  016652             .WORD   .$.
9074 025050                     MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9075         000046             MICPC=MICPC+1
```

```
9076  025050 016652                .WORD   .$,
9077  025052                       MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
9078         000047                MICPC=MICPC+1
9079  025052 016652                .WORD   .$.
9080  025054                       MOVE    # 252,MEM MARINC        ;RESULT WITH C BIT SET.
9081         000050                MICPC=MICPC+1
9082  025054 016652                .WORD   .$,
9083  025056                       MOVE    # 7,SPAD <7>     ;SET THE COUNT.
9084         000051                MICPC=MICPC+1
9085  025056 003007                .WORD   .$.
9086
9087  025060                       MOVE    # 0,MLR          ;MAR+0.
9088         000052                MICPC=MICPC+1
9089  025060 010000                .WORD   .$.
9090
9091  025062                 2$:   MOVE    # 0,BREG                   ;
9092         000053                MICPC=MICPC+1
9093  025062 000400                .WORD   .$.
9094  025064                       $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
9095         000054                MICPC=MICPC+1
9096  025064 060401                .WORD   .$.!.DD
9097  025066                       MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
9098         000055                MICPC=MICPC+1
9099  025066 057220                .WORD   .$.
9100  025070                       MOVE    SPAD <0>,BR.SP,MARINC   ;
9101         000056                MICPC=MICPC+1
9102  025070 077600                .WORD   .$.
9103  025072                       $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
9104
9105
9106  025072                       SUB2C   SPAD <0>,MEM,NOP
9107         000057                MICPC=MICPC+1
9108  025072 040360                .WORD   .$.
9109  025074                       BZ      3$
9110         000060                MICPC=MICPC+1
9111  025074 101473                .WORD   .$.
9112  025076                       MOVE    MEM,OUT1 <4>     ;GOOD DATA
9113         000061                MICPC=MICPC+1
9114  025076 041224                .WORD   .$.
9115  025100                       MOVE    BREG,OUT1 <5>    ;BAD DATA.
9116         000062                MICPC=MICPC+1
9117  025100 061225                .WORD   .$,
9118  025102                       MOVE    # 15,BREG        ;SET TYPE OF ERROR.
9119         000063                MICPC=MICPC+1
9120  025102 000415                .WORD   .$.
9121  025104                       MOVE    BREG,OUT1 <3>    ;SET TYPE OF ERROR.
9122         000064                MICPC=MICPC+1
9123  025104 061223                .WORD   .$.
9124  025106                       MOVE    # 11,BREG        ;LOAD FUNCTION CODE...
9125         000065                MICPC=MICPC+1
9126  025106 000411                .WORD   .$.
9127  025110                       MOVE    BREG,OUT1 <CSR7>         ;LOAD IT...
9128         000066                MICPC=MICPC+1
9129  025110 061227                .WORD   .$.
9130  025112                       CALL    ERDR             ;ALU SEL B & SEL A ERROR...
9131  025112                       MOVE    # <MICPC+3>,BREG
```

```
9132                    000067                  MICPC=MICPC+1
9133    025112          000471                  .WORD   .$.
9134    025114                                  SBR     ERROR
9135                    000070                  MICPC=MICPC+1
9136    025114          104400                  .WORD   .$.
9137    025116                                  MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
9138                    000071                  MICPC=MICPC+1
9139    025116          070204                  .WORD   .$.
9140    025120                                  SBR     2$              ;LOOP ON ERROR...
9141                    000072                  MICPC=MICPC+1
9142    025120          100453                  .WORD   .$.
9143    025122                          3$:     CALL    SCP1
9144    025122                                  MOVE    # <MICPC+3>,BREG
9145                    000073                  MICPC=MICPC+1
9146    025122          000475                  .WORD   .$.
9147    025124                                  SBR     SCP1
9148                    000074                  MICPC=MICPC+1
9149    025124          104427                  .WORD   .$.
9150    025126                                  MOVE    SPAD <4>,MLR    ;
9151                    000075                  MICPC=MICPC+1
9152    025126          070204                  .WORD   .$.
9153    025130                                  SBR     2$              ; SCOPE THE DATA....
9154                    000076                  MICPC=MICPC+1
9155    025130          100453                  .WORD   .$.
9156    025132                          6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
9157                    000077                  MICPC=MICPC+1
9158    025132          070204                  .WORD   .$.
9159    025134                                  MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
9160                    000100                  MICPC=MICPC+1
9161    025134          057220                  .WORD   .$.
9162    025136                                  MOVE    # 377,BREG      ;
9163                    000101                  MICPC=MICPC+1
9164    025136          000777                  .WORD   .$.
9165    025140                                  SPAD    SPAD <2>,BREG   ;SET C BIT...
9166                    000102                  MICPC=MICPC+1
9167    025140          060402                  .WORD   .$.!.DD
9168    025142                                  MOVE    SPAD <0>,BR.SP,MARINC           ;SP 0 & BR = SEL B & SEL A
9169                    000103                  MICPC=MICPC+1
9170    025142          077600                  .WORD   .$.
9171    025144                                  MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
9172                    000104                  MICPC=MICPC+1
9173    025144          057223                  .WORD   .$.
9174    025146                                  SIFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
9175
9176
9177    025146                                  SUB2C   SPAD <0>,MEM,NOP
9178                    000105                  MICPC=MICPC+1
9179    025146          040360                  .WORD   .$.
9180    025150                                  BZ      9$
9181                    000106                  MICPC=MICPC+1
9182    025150          101521                  .WORD   .$.
9183    025152                                  MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
9184                    000107                  MICPC=MICPC+1
9185    025152          041224                  .WORD   .$.
9186    025154                                  MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
9187                    000110                  MICPC=MICPC+1
```

```
 9188  025154  061225                    .WORD   .$.
 9189  025156                            MOVE    # 23,BREG               ;ERROR TYPE...
 9190          000111                    MICPC=MICPC+1
 9191  025156  000423                    .WORD   .$.
 9192  025160                            MOVE    BREG,OUT1 <CSR3>        ;
 9193          000112                    MICPC=MICPC+1
 9194  025160  061223                    .WORD   .$.
 9195  025162                            MOVE    # 11,BREG               ;ALU FUNCTION CODE.
 9196          000113                    MICPC=MICPC+1
 9197  025162  000411                    .WORD   .$.
 9198  025164                            MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
 9199          000114                    MICPC=MICPC+1
 9200  025164  061227                    .WORD   .$.
 9201  025166                            CALL    ERROR   ;REPORT ERROR...
 9202  025166                            MOVE    # <MICPC+3>,BREG
 9203          000115                    MICPC=MICPC+1
 9204  025166  000517                    .WORD   .$.
 9205  025170                            SBR     ERROR
 9206          000116                    MICPC=MICPC+1
 9207  025170  104400                    .WORD   .$.
 9208  025172                            MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
 9209          000117                    MICPC=MICPC+1
 9210  025172  070204                    .WORD   .$.
 9211  025174                            SBR     6$                      ;LOOP ON ERROR...
 9212          000120                    MICPC=MICPC+1
 9213  025174  100477                    .WORD   .$.
 9214  025176                    9$:     CALL    SCP1            ;SCOPE THE ERROR...
 9215  025176                            MOVE    # <MICPC+3>,BREG
 9216          000121                    MICPC=MICPC+1
 9217  025176  000523                    .WORD   .$.
 9218  025200                            SBR     SCP1
 9219          000122                    MICPC=MICPC+1
 9220  025200  104427                    .WORD   .$.
 9221  025202                            MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
 9222          000123                    MICPC=MICPC+1
 9223  025202  070204                    .WORD   .$.
 9224  025204                            SBR     6$              ;SCOPE THE DATA...
 9225          000124                    MICPC=MICPC+1
 9226  025204  100477                    .WORD   .$.
 9227  025206                            MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
 9228          000125                    MICPC=MICPC+1
 9229  025206  000404                    .WORD   .$.
 9230  025210                            SADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
 9231  025210  077004                    .WORD   .$.!MARINC!.DD
 9232  025212                            SDEC    SPAD <7>                ;IS IT DONE??
 9233          000126                    MICPC=MICPC+1
 9234  025212  063167            .WORD   .$.!.DSP
 9235  025214                            BZ      4$                      ;YES, SCOPE THE TEST.
 9236          000127                    MICPC=MICPC+1
 9237  025214  101532                    .WORD   .$.
 9238  025216                            SBR     2$                      ;DO, THE NEXT.
 9239          000130                    MICPC=MICPC+1
 9240  025216  100453                    .WORD   .$.
 9241  025220                    4$:     CALL    SCPE            ;SCOPE THE TEST...
 9242  025220                            MOVE    # <MICPC+3>,BREG
 9243          000131                    MICPC=MICPC+1
```

```
                                        .WORD   .$
                                        SBR     SCPE
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                        SBR     1$                      ;DO THE NEXT ITERATION...
                                        MICPC=MICPC+1
                                        .WORD   .$.
                                SALUT1  0,<A OR NOTB>,LORN,-1,-1,-1,-1,0,0,-1,-1,-1,-1,252,252,125,125,-1,-1,<A OR NOTB>
                                        SXZ


                                        ;*************************** TEST 44 ****************************
                                        ;#ALU TEST
                                        ;#TEST OF ALU FUNCTION A OR NOTB WITH C BIT CLEARED.
                                        ;#TEST OF ALU FUNCTION A OR NOTB WITH C BIT SET.
                                        ;#ALU FUNCTION (A OR NOTB)
                                        ;#LOAD MAIN MEMORY 16 WORDS OF DATA.
                                        ;#PERFORM THE FUNCTION, VERIFY THE RESULTS..
                                SXZ     ;;************************************************************************
                                STSTN
                                        ;  TEST 44
                                        ;-------------
025226 012737 000044 001202     TST44:  MOV     #44,STSTNM              ; LOAD THE NO. OF THIS TEST
025234 012737 025554 001442             MOV     #TST45,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
025242 004737 035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
025246 025262
025250 104022                           MCT44
                                        ERROR   22                      ;TIME OUT ERROR...
025252 012706 001200                    MOV     #STACK,SP               ;RESET STACK...
025256 000177 154160                    JMP     @NEXT                   ;GO TO NEXT TEST...
025262                          MCT44:
                                1$:     MOVE    # 0,MLR          ;SET MAR<-LO.
000000                                  MICPC=MICPC+1
025262 010000                           .WORD   .$.
025264                                  MOVE    # 0,MPR          ;SET MAR<-HI.
000001                                  MICPC=MICPC+1
025264 004000                           .WORD   .$.
025266                                  MOVE    # 0,BREG                 ;
000002                                  MICPC=MICPC+1
025266 000400                           .WORD   .$.
025270                                  MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
000003                                  MICPC=MICPC+1
025270 063236                           .WORD   .$.
025272                                  MOVE    BREG,SPAD <0>   ;
000004                                  MICPC=MICPC+1
025272 063220                           .WORD   .$.
025274                                  MOVE    BREG,SPAD <1>   ;
000005                                  MICPC=MICPC+1
025274 063221                           .WORD   .$.
025276                                  MOVE    BREG,SPAD <2>   ;
000006                                  MICPC=MICPC+1
025276 063222                           .WORD   .$.
025300                                  $DEC    SPAD <2>                ;
000007                                  MICPC=MICPC+1
```

```
                                .WORD   $,!.DSP
                                        MOVE    BREG,SPAD <4>           ;
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # -1,MEM MARINC
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # -1,MEM MARINC
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # 0,MEM MARINC
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # -1,MEM MARINC
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
                                        MICPC=MICPC+1
                                .WORD   .S.
                                        MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
```

```
 9356                   000032                  MICPC=MICPC+1
 9357  025346  016525                           .WORD   .S.
 9358  025350                                   MOVE    # -1,MEM MARINC
 9359                   000033                  MICPC=MICPC+1
 9360  025350  016777                           .WORD   .S.
 9361  025352                                   MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
 9362                   000034                  MICPC=MICPC+1
 9363  025352  016777                           .WORD   .S.
 9364  025354                                   MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
 9365                   000035                  MICPC=MICPC+1
 9366  025354  016652                           .WORD   .S.
 9367  025356                                   MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
 9368                   000036                  MICPC=MICPC+1
 9369  025356  016525                           .WORD   .S.
 9370  025360                                   MOVE    # 252,MEM MARINC
 9371                   000037                  MICPC=MICPC+1
 9372  025360  016652                           .WORD   .S.
 9373  025362                                   MOVE    # 252,MEM MARINC         ;RESULT WITH C BIT SET.
 9374                   000040                  MICPC=MICPC+1
 9375  025362  016652                           .WORD   .S.
 9376  025364                                   MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
 9377                   000041                  MICPC=MICPC+1
 9378  025364  016525                           .WORD   .S.
 9379  025366                                   MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY
 9380                   000042                  MICPC=MICPC+1
 9381  025366  016652                           .WORD   .S.
 9382  025370                                   MOVE    # 125,MEM MARINC
 9383                   000043                  MICPC=MICPC+1
 9384  025370  016525                           .WORD   .S.
 9385  025372                                   MOVE    # 125,MEM MARINC         ;RESULT WITH C BIT SET.
 9386                   000044                  MICPC=MICPC+1
 9387  025372  016525                           .WORD   .S.
 9388  025374                                   MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
 9389                   000045                  MICPC=MICPC+1
 9390  025374  016652                           .WORD   .S.
 9391  025376                                   MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
 9392                   000046                  MICPC=MICPC+1
 9393  025376  016652                           .WORD   .S.
 9394  025400                                   MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
 9395                   000047                  MICPC=MICPC+1
 9396  025400  016777                           .WORD   .S.
 9397  025402                                   MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
 9398                   000050                  MICPC=MICPC+1
 9399  025402  016777                           .WORD   .S.
 9400  025404                                   MOVE    # 7,SPAD <7>     ;SET THE COUNT.
 9401                   000051                  MICPC=MICPC+1
 9402  025404  003007                           .WORD   .S.
 9403
 9404  025406                                   MOVE    # 0,MLR          ;MAR=0.
 9405                   000052                  MICPC=MICPC+1
 9406  025406  010000                           .WORD   .S.
 9407
 9408  025410                           2$:     MOVE    # 0,BREG                 ;
 9409                   000053                  MICPC=MICPC+1
 9410  025410  000400                           .WORD   .S.
 9411  025412                                   $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
```

```
                                        MICPC=MICPC+1
000054
025412  060401                          .WORD   .$.!.DD
025414                                  MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
                                        MICPC=MICPC+1
000055
025414  057220                          .WORD   .$.
025416                                  LORM    MEM,SPAD <0>,BR.SP,MARINC       ;SPAD  <0>:=SFUNC.
                                        MICPC=MICPC+1
000056
025416  057640                          .WORD   .$.
025420                                  SIFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.


025420                                  SUB2C   SPAD <0>,MEM,NOP
000057                                  MICPC=MICPC+1
025420  040360                          .WORD   .$.
025422                                  BZ      3$
                                        MICPC=MICPC+1
000060
025422  101473                          .WORD   .$.
025424                                  MOVE    MEM,OUT1 <4>    ;GOOD DATA
                                        MICPC=MICPC+1
000061
025424  041224                          .WORD   .$.
025426                                  MOVE    BREG,OUT1 <5>   ;BAD DATA.
                                        MICPC=MICPC+1
000062
025426  061225                          .WORD   .$.
025430                                  MOVE    # 15,BREG       ;SET TYPE OF ERROR.
                                        MICPC=MICPC+1
000063
025430  000415                          .WORD   .$.
025432                                  MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
                                        MICPC=MICPC+1
0000
025432  061223                          .WORD   .$.
025434                                  MOVE    # 12,BREG       ;LOAD FUNCTION CODE...
                                        MICPC=MICPC+1
000065
025434  000412                          .WORD   .$.
025436                                  MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
                                        MICPC=MICPC+1
000066
025436  061227                          .WORD   .$.
025440                                  CALL    ERROR           ;ALU A OR NOTB ERROR...
025440                                  MOVE    # <MICPC+3>,BREG
                                        MICPC=MICPC+1
000067
025440  000471                          .WORD   .$.
025442                                  SBR     ERROR
                                        MICPC=MICPC+1
000070
025442  104400                          .WORD   .$.
025444                                  MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
                                        MICPC=MICPC+1
000071
025444  070204                          .WORD   .$.
025446                                  SBR     2$              ;LOOP ON ERROR...
                                        MICPC=MICPC+1
000072
025446  100453                          .WORD   .$.
025450                          3$:     CALL    SCP1
025450                                  MOVE    # <MICPC+3>,BREG
                                        MICPC=MICPC+1
000073
025450  000475                          .WORD   .$.
025452                                  SBR     SCP1
                                        MICPC=MICPC+1
000074
025452  104427                          .WORD   .$.
025454                                  MOVE    SPAD <4>,MLR    ;
```

```
                                        MICPC=MICPC+1
025454  070204                          .WORD   .$.
025456                                  $BR     23                      ; SCOPE THE DATA....
                                        MICPC=MICPC+1
025456  100453                          .WORD   .$.
025460                          65:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
                                        MICPC=MICPC+1
025460  070204                          .WORD   .$.
025462                                  MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
                                        MICPC=MICPC+1
025462  057220                          .WORD   .$.
025464                                  MOVE    # 377,BREG      ;
                                        MICPC=MICPC+1
025464  000777                          .WORD   .$.
025466                                  $ADD    SPAD <2>,BREG   ;SET C BIT...
                                        MICPC=MICPC+1
025466  060402                          .WORD   .$.!.DD
025470                                  LORM    MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = A OR NOTB
                                        MICPC=MICPC+1
025470  057640                          .WORD   .$.
025472                                  MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
                                        MICPC=MICPC+1
025472  057223                          .WORD   .$.
025474                                  $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...


025474                                  SUB2C   SPAD <0>,MEM,NOP
                                        MICPC=MICPC+1
025474  040360                          .WORD   .$.
025476                                  BZ      9$
                                        MICPC=MICPC+1
025476  101521                          .WORD   .$.
025500                                  MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
                                        MICPC=MICPC+1
025500  041224                          .WORD   .$.
025502                                  MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
                                        MICPC=MICPC+1
025502  061225                          .WORD   .$.
025504                                  MOVE    # 23,BREG               ;ERROR TYPE...
                                        MICPC=MICPC+1
025504  000423                          .WORD   .$.
025506                                  MOVE    BREG,OUT1 <CSR3>        ;
                                        MICPC=MICPC+1
025506  061223                          .WORD   .$.
025510                                  MOVE    # 12,BREG               ;ALU FUNCTION CODE.
                                        MICPC=MICPC+1
025510  000412                          .WORD   .$.
025512                                  MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
                                        MICPC=MICPC+1
025512  061227                          .WORD   .$.
025514                                  CALL    EROR    ;REPORT ERROR...
025514                                  MOVE    # <MICPC+3>,BREG
                                        MICPC=MICPC+1
025514  000517                          .WORD   .$.
025516                                  $BR     EROR
                                        MICPC=MICPC+1
```

```
025516 104400                          .WORD   .$.
025520                                 MOVE    SPAD (4),MLR            ;RESTORE DATA POINTER.
000117                                 MICPC=MICPC+1
025520 070204                          .WORD   .$.
025522                                 SBR     6$                      ;LOOP ON ERROR...
000120                                 MICPC=MICPC+1
025522 100477                          .WORD   .$.
025524                          9$:    CALL    SCP1            ;SCOPE THE ERROR...
025524                                 MOVE    # (MICPC+3),BREG
000121                                 MICPC=MICPC+1
025524 000523                          .WORD   .$.
025526                                 SBR     SCP1
000122                                 MICPC=MICPC+1
025526 104427                          .WORD   .$.
025530                                 MOVE    SPAD (4),MLR    ;RESTORE DATA POINTER...
000123                                 MICPC=MICPC+1
025530 070204                          .WORD   .$.
025532                                 SBR     6$                      ;SCOPE THE DATA...
000124                                 MICPC=MICPC+1
025532 100477                          .WORD   .$.
025534                                 MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
000125                                 MICPC=MICPC+1
025534 000404                          .WORD   .$.
025536                                 SADD    BREG,SPAD (4),MARINC    ;ALSO DATA POINTER.
025536 077004                          .WORD   .$.!MARINC!.DD
025540                                 SDEC    SPAD (7)                ;IS IT DONE??
000126                                 MICPC=MICPC+1
025540 063167                  .WORD   .$.!.DSP
025542                                 BZ      4$                      ;YES, SCOPE THE TEST.
000127                                 MICPC=MICPC+1
025542 101532                          .WORD   .$.
025544                                 SBR     2$                      ;DO, THE NEXT.
000130                                 MICPC=MICPC+1
025544 100453                          .WORD   .$.
025546                          4$:    CALL    SCPE            ;SCOPE THE TEST...
025546                                 MOVE    # (MICPC+3),BREG
000131                                 MICPC=MICPC+1
025546 000533                          .WORD   .$.
025550                                 SBR     SCPE
000132                                 MICPC=MICPC+1
025550 104454                          .WORD   .$.
025552                                 SBR     1$                      ;DO THE NEXT ITERATION...
000133                                 MICPC=MICPC+1
025552 100400                          .WORD   .$.
025554                          SALUT1 0,<A AND B>,AND,0,0,0,0,0,0,-1,-1,125,125,0,0,0,0,252,252,<A AND B>,1,13
025554                          $XZ


                                ;************************** TEST 45 ****************************
                                ;*ALU TEST
                                ;*TEST OF ALU FUNCTION A AND B WITH C BIT CLEARED.
                                ;*TEST OF ALU FUNCTION A AND B WITH C BIT SET.
                                ;*ALU FUNCTION (A AND B)
                                ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
                                ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
025554                          $XZ
```

```
                                        ;:*****************************************************************
                                $TSTN
                                ;  TEST 45
                                ;-------------
025554 012737 000045 001202 TST45:  MOV     #45,$TSTNM          ; LOAD THE NO. OF THIS TEST
025562 012737 026102 001442         MOV     #TST46,NEXT         ; POINT TO THE START OF NEXT TEST.
                                                        ;R1 CONTAINS BASE KMC11 ADDRESS
025570 004737 035536                JSR     PC,LDVRWT           ;LOAD-VERIFY-WAIT.
025574 025610                       MCT45
025576 104022                       ERROR   22                  ;TIME OUT ERROR...
025600 012706 001200                MOV     #STACK,SP           ;RESET STACK...
025604 000177 153632                JMP     @NEXT               ;GO TO NEXT TEST...
025610                          MCT45:
025610                          1$:     MOVE    # 0,MLR          ;SET MAR<-LO.
000000                                  MICPC=MICPC+1
025610 010000                           .WORD   .$.
025612                                  MOVE    # 0,MPR          ;SET MAR<-HI.
000001                                  MICPC=MICPC+1
025612 004000                           .WORD   .$.
025614                                  MOVE    # 0,BREG          ;
000002                                  MICPC=MICPC+1
025614 000400                           .WORD   .$.
025616                                  MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS...
000003                                  MICPC=MICPC+1
025616 063236                           .WORD   .$.
025620                                  MOVE    BREG,SPAD (0)   ;
000004                                  MICPC=MICPC+1
025620 063220                           .WORD   .$.
025622                                  MOVE    BREG,SPAD (1)   ;
000005                                  MICPC=MICPC+1
025622 063221                           .WORD   .$.
025624                                  MOVE    BREG,SPAD (2)   ;
000006                                  MICPC=MICPC+1
025624 063222                           .WORD   .$.
025626                                  SDEC    SPAD (2)                ;
000007                                  MICPC=MICPC+1
025626 063162                   .WORD   .$.!.DSP
025630                                  MOVE    BREG,SPAD (4)           ;
000010                                  MICPC=MICPC+1
025630 063224                           .WORD   .$.
025632                                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
000011                                  MICPC=MICPC+1
025632 016400                           .WORD   .$.
025634                                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
000012                                  MICPC=MICPC+1
025634 016400                           .WORD   .$.
025636                                  MOVE    # 0,MEM MARINC
000013                                  MICPC=MICPC+1
025636 016400                           .WORD   .$.
025640                                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
000014                                  MICPC=MICPC+1
025640 016400                           .WORD   .$.
025642                                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
000015                                  MICPC=MICPC+1
025642 016777                           .WORD   .$.
```

```
9636  025644                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
9637          000016          MICPC=MICPC+1
9638  025644  016400          .WORD   .S.
9639  025646                  MOVE    # 0,MEM MARINC
9640          000017          MICPC=MICPC+1
9641  025646  016400          .WORD   .S.
9642  025650                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
9643          000020          MICPC=MICPC+1
9644  025650  016400          .WORD   .S.
9645  025652                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
9646          000021          MICPC=MICPC+1
9647  025652  016400          .WORD   .S.
9648  025654                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
9649          000022          MICPC=MICPC+1
9650  025654  016777          .WORD   .S.
9651  025656                  MOVE    # 0,MEM MARINC
9652          000023          MICPC=MICPC+1
9653  025656  016400          .WORD   .S.
9654  025660                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
9655          000024          MICPC=MICPC+1
9656  025660  016400          .WORD   .S.
9657  025662                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
9658          000025          MICPC=MICPC+1
9659  025662  016777          .WORD   .S.
9660  025664                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
9661          000026          MICPC=MICPC+1
9662  025664  016777          .WORD   .S.
9663  025666                  MOVE    # -1,MEM MARINC
9664          000027          MICPC=MICPC+1
9665  025666  016777          .WORD   .S.
9666  025670                  MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
9667          000030          MICPC=MICPC+1
9668  025670  016777          .WORD   .S.
9669  025672                  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9670          000031          MICPC=MICPC+1
9671  025672  016525          .WORD   .S.
9672  025674                  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9673          000032          MICPC=MICPC+1
9674  025674  016525          .WORD   .S.
9675  025676                  MOVE    # 125,MEM MARINC
9676          000033          MICPC=MICPC+1
9677  025676  016525          .WORD   .S.
9678  025700                  MOVE    # 125,MEM MARINC         ;RESULT WITH C BIT SET.
9679          000034          MICPC=MICPC+1
9680  025700  016525          .WORD   .S.
9681  025702                  MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9682          000035          MICPC=MICPC+1
9683  025702  016652          .WORD   .S.
9684  025704                  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9685          000036          MICPC=MICPC+1
9686  025704  016525          .WORD   .S.
9687  025706                  MOVE    # 0,MEM MARINC
9688          000037          MICPC=MICPC+1
9689  025706  016400          .WORD   .S.
9690  025710                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
9691          000040          MICPC=MICPC+1
```

```
9692  025710  016400                .WORD   .S.
9693  025712                        MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
9694          000041                MICPC=MICPC+1
9695  025712  016525                .WORD   .S.
9696  025714                        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
9697          000042                MICPC=MICPC+1
9698  025714  016652                .WORD   .S.
9699  025716                        MOVE    # 0,MEM MARINC
9700          000043                MICPC=MICPC+1
9701  025716  016400                .WORD   .S.
9702  025720                        MOVE    # 0,MEM MARINC   ;RESULT WITH C BIT SET.
9703          000044                MICPC=MICPC+1
9704  025720  016400                .WORD   .S.
9705  025722                        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9706          000045                MICPC=MICPC+1
9707  025722  016652                .WORD   .S.
9708  025724                        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9709          000046                MICPC=MICPC+1
9710  025724  016652                .WORD   .S.
9711  025726                        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
9712          000047                MICPC=MICPC+1
9713  025726  016652                .WORD   .S.
9714  025730                        MOVE    # 252,MEM MARINC         ;RESULT WITH C BIT SET.
9715          000050                MICPC=MICPC+1
9716  025730  016652                .WORD   .S.
9717  025732                        MOVE    # 7,SPAD <7>     ;SET THE COUNT.
9718          000051                MICPC=MICPC+1
9719  025732  003007                .WORD   .S.
9720
9721  025734                        MOVE    # 0,MLR          ;MAR+0.
9722          000052                MICPC=MICPC+1
9723  025734  010000                .WORD   .S.
9724
9725  025736                2$:     MOVE    # 0,BREG                  ;
9726          000053                MICPC=MICPC+1
9727  025736  000400                .WORD   .S.
9728  025740                        $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
9729          000054                MICPC=MICPC+1
9730  025740  060401                .WORD   .S.!.DD
9731  025742                        MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
9732          000055                MICPC=MICPC+1
9733  025742  057220                .WORD   .S.
9734  025744                        AND     MEM,SPAD <0>,BR.SP,MARINC       ;SPAD  <0>:=$FUNC.
9735          000056                MICPC=MICPC+1
9736  025744  057660                .WORD   .S.
9737  025746                        $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
9738
9739
9740  025746                        SUB2C   SPAD <0>,MEM,NOP
9741          000057                MICPC=MICPC+1
9742  025746  040360                .WORD   .S.
9743  025750                        BZ      3$
9744          000060                MICPC=MICPC+1
9745  025750  101473                .WORD   .S.
9746  025752                        MOVE    MEM,OUT1 <4>    ;GOOD DATA
9747          000061                MICPC=MICPC+1
```

```
9748  025752  041224                    .WORD   .$.
9749  025754                            MOVE    BREG,OUT1 <5>   ;BAD DATA.
9750          000062                    MICPC=MICPC+1
9751  025754  061225                    .WORD   .$.
9752  025756                            MOVE    # 15,BREG       ;SET TYPE OF ERROR.
9753          000063                    MICPC=MICPC+1
9754  025756  000415                    .WORD   .$.
9755  025760                            MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
9756          000064                    MICPC=MICPC+1
9757  025760  061223                    .WORD   .$.
9758  025762                            MOVE    # 13,BREG       ;LOAD FUNCTION CODE...
9759          000065                    MICPC=MICPC+1
9760  025762  000413                    .WORD   .$.
9761  025764                            MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
9762          000066                    MICPC=MICPC+1
9763  025764  061227                    .WORD   .$.
9764  025766                            CALL    ERROR           ;ALU A AND B ERROR...
9765  025766                            MOVE    # <MICPC+3>,BREG
9766          000067                    MICPC=MICPC+1
9767  025766  000471                    .WORD   .$.
9768  025770                            SBR     ERROR
9769          000070                    MICPC=MICPC+1
9770  025770  104400                    .WORD   .$.
9771  025772                            MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
9772          000071                    MICPC=MICPC+1
9773  025772  070204                    .WORD   .$.
9774  025774                            SBR     2$              ;LOOP ON ERROR...
9775          000072                    MICPC=MICPC+1
9776  025774  100453                    .WORD   .$.
9777  025776                    3$:     CALL    SCP1
9778  025776                            MOVE    # <MICPC+3>,BREG
9779          000073                    MICPC=MICPC+1
9780  025776  000475                    .WORD   .$.
9781  026000                            SBR     SCP1
9782          000074                    MICPC=MICPC+1
9783  026000  104427                    .WORD   .$.
9784  026002                            MOVE    SPAD <4>,MLR    ;
9785          000075                    MICPC=MICPC+1
9786  026002  070204                    .WORD   .$.
9787  026004                            SBR     2$              ; SCOPE THE DATA....
9788          000076                    MICPC=MICPC+1
9789  026004  100453                    .WORD   .$.
9790  026006                    6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
9791          000077                    MICPC=MICPC+1
9792  026006  070204                    .WORD   .$.
9793  026010                            MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
9794          000100                    MICPC=MICPC+1
9795  026010  057220                    .WORD   .$.
9796  026012                            MOVE    # 377,BREG      ;
9797          000101                    MICPC=MICPC+1
9798  026012  000777                    .WORD   .$.
9799  026014                            SADD    SPAD <2>,BREG   ;SET C BIT...
9800          000102                    MICPC=MICPC+1
9801  026014  060402                    .WORD   .$.!.DD
9802  026016                            AND     MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = A AND B
9803          000103                    MICPC=MICPC+1
```

```
 9804  026016  057660                    .WORD   $
 9805  026020                            MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
 9806          000104                    MICPC=MICPC+1
 9807  026020  057223                    .WORD   $
 9808  026022                            $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
 9809
 9810
 9811  026022                            SUB2C   SPAD <0>,MEM,NOP
 9812          000105                    MICPC=MICPC+1
 9813  026022  040360                    .WORD   .$.
 9814  026024                            BZ      9$
 9815          000106                    MICPC=MICPC+1
 9816  026024  101521                    .WORD   .$.
 9817  026026                            MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
 9818          000107                    MICPC=MICPC+1
 9819  026026  041224                    .WORD   .$.
 9820  026030                            MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
 9821          000110                    MICPC=MICPC+1
 9822  026030  061225                    .WORD   .$.
 9823  026032                            MOVE    # 23,BREG               ;ERROR TYPE...
 9824          000111                    MICPC=MICPC+1
 9825  026032  000423                    .WORD   .$.
 9826  026034                            MOVE    BREG,OUT1 <CSR3>        ;
 9827          000112                    MICPC=MICPC+1
 9828  026034  061223                    .WORD   .$.
 9829  026036                            MOVE    # 13,BREG               ;ALU FUNCTION CODE.
 9830          000113                    MICPC=MICPC+1
 9831  026036  000413                    .WORD   .$.
 9832  026040                            MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
 9833          000114                    MICPC=MICPC+1
 9834  026040  061227                    .WORD   .$.
 9835  026042                            CALL    ERROR   ;REPORT ERROR...
 9836  026042                            MOVE    # <MICPC+3>,BREG
 9837          000115                    MICPC=MICPC+1
 9838  026042  000517                    .WORD   .$.
 9839  026044                            SBR     ERROR
 9840          000116                    MICPC=MICPC+1
 9841  026044  104400                    .WORD   .$.
 9842  026046                            MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
 9843          000117                    MICPC=MICPC+1
 9844  026046  070204                    .WORD   .$.
 9845  026050                            SBR     6$                      ;LOOP ON ERROR...
 9846          000120                    MICPC=MICPC+1
 9847  026050  100477                    .WORD   .$.
 9848  026052                    9$:     CALL    SCP1                    ;SCOPE THE ERROR...
 9849  026052                            MOVE    # <MICPC+3>,BREG
 9850          000121                    MICPC=MICPC+1
 9851  026052  000523                    .WORD   .$.
 9852  026054                            SBR     SCP1
 9853          000122                    MICPC=MICPC+1
 9854  026054  104427                    .WORD   .$.
 9855  026056                            MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
 9856          000123                    MICPC=MICPC+1
 9857  026056  070204                    .WORD   .$.
 9858  026060                            SBR     6$                      ;SCOPE THE DATA...
 9859          000124                    MICPC=MICPC+1
```

```
026060 100477                          .WORD   .$
026062                                 MOVE    #4,BREG                 ;UPDATE BACKGROUND POINTER.
       000125                          MICPC=MICPC+1
026062 000404                          .WORD   .$.
026064                                 SADD    BREG,SPAD<4>,MARINC     ;ALSO DATA POINTER.
026064 077004                          .WORD   .$.!MARINC!.DD
026066                                 SDEC    SPAD<7>                 ;IS IT DONE??
       000126                          MICPC=MICPC+1
026066 063167                  .WORD   .$.!.DSP
026070                                 BZ      4$                      ;YES, SCOPE THE TEST.
       000127                          MICPC=MICPC+1
026070 101532                          .WORD   .$.
026072                                 SBR     2$                      ;DO, THE NEXT.
       000130                          MICPC=MICPC+1
026072 100453                          .WORD   .$.
026074                         4$:     CALL    SCPE                    ;SCOPE THE TEST...
026074                                 MOVE    #<MICPC+3>,BREG
       000131                          MICPC=MICPC+1
026074 000533                          .WORD   .$.
026076                                 SBR     SCPE
       000132                          MICPC=MICPC+1
026076 104454                          .WORD   .$.
026100                                 SBR     1$                      ;DO THE NEXT ITERATION...
       000133                          MICPC=MICPC+1
026100 100400                          .WORD   .$.
026102                         SALUT1  0,<A OR B>,OR,0,0,-1,-1,-1,-1,-1,-1,125,125,-1,-1,-1,-1,252,252,<A OR B>,1,14
026102                         SXZ


                               ;************************** TEST 46 ****************************
                               ;*ALU TEST
                               ;*TEST OF ALU FUNCTION A OR B WITH C BIT CLEARED.
                               ;*TEST OF ALU FUNCTION A OR B WITH C BIT SET.
                               ;*ALU FUNCTION (A OR B)
                               ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
                               ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
026102                         SXZ
                               ;;*************************************************************
026102                         STSTN

                               ;  TEST 46
                               ;--------------
026102 012737 000046 001202    TST46:  MOV     #46,STSTNM              ; LOAD THE NO. OF THIS TEST
026110 012737 026430 001442            MOV     #TST47,NEXT             ; POINT TO THE START OF NEXT TEST.
                                                               ;R1 CONTAINS BASE KMC11 ADDRESS
026116 004737 035536                   JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
026122 026136                          MCT46
026124 104022                          ERROR   22                      ;TIME OUT ERROR...
026126 012706 001200                   MOV     #STACK,SP               ;RESET STACK...
026132 000177 153304                   JMP     @NEXT                   ;GO TO NEXT TEST...
026136                         MCT46:
026136                         1$:     MOVE    # 0,MLR          ;SET MAR<-LO.
       000000                          MICPC=MICPC+1
026136 010000                          .WORD   .$.
026140                                 MOVE    # 0,MPR          ;SET MAR<-HI.
       000001                          MICPC=MICPC+1
```

```
9916  026140  004000                    .WORD   .$
9917  026142                            MOVE    # 0,BREG                ;
9918          000002                    MICPC=MICPC+1
9919  026142  000400                    .WORD   .$
9920  026144                            MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
9921          000003                    MICPC=MICPC+1
9922  026144  063236                    .WORD   .$
9923  026146                            MOVE    BREG,SPAD <0>   ;
9924          000004                    MICPC=MICPC+1
9925  026146  063220                    .WORD   .$
9926  026150                            MOVE    BREG,SPAD <1>   ;
9927          000005                    MICPC=MICPC+1
9928  026150  063221                    .WORD   .$
9929  026152                            MOVE    BREG,SPAD <2>   ;
9930          000006                    MICPC=MICPC+1
9931  026152  063222                    .WORD   .$
9932  026154                            SDEC    SPAD <2>                ;
9933          000007                    MICPC=MICPC+1
9934  026154  063162            .WORD   .$.!.DSP
9935  026156                            MOVE    BREG,SPAD <4>           ;
9936          000010                    MICPC=MICPC+1
9937  026156  063224                    .WORD   .$
9938  026160                            MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
9939          000011                    MICPC=MICPC+1
9940  026160  016400                    .WORD   .$
9941  026162                            MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
9942          000012                    MICPC=MICPC+1
9943  026162  016400                    .WORD   .$
9944  026164                            MOVE    # 0,MEM MARINC
9945          000013                    MICPC=MICPC+1
9946  026164  016400                    .WORD   .$
9947  026166                            MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
9948          000014                    MICPC=MICPC+1
9949  026166  016400                    .WORD   .$
9950  026170                            MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
9951          000015                    MICPC=MICPC+1
9952  026170  016777                    .WORD   .$
9953  026172                            MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
9954          000016                    MICPC=MICPC+1
9955  026172  016400                    .WORD   .$
9956  026174                            MOVE    # -1,MEM MARINC
9957          000017                    MICPC=MICPC+1
9958  026174  016777                    .WORD   .$
9959  026176                            MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
9960          000020                    MICPC=MICPC+1
9961  026176  016777                    .WORD   .$
9962  026200                            MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
9963          000021                    MICPC=MICPC+1
9964  026200  016400                    .WORD   .$
9965  026202                            MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
9966          000022                    MICPC=MICPC+1
9967  026202  016777                    .WORD   .$
9968  026204                            MOVE    # -1,MEM MARINC
9969          000023                    MICPC=MICPC+1
9970  026204  016777                    .WORD   .$
9971  026206                            MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
```

```
 9972            000024
 9973  026206    016777          MICPC=MICPC+1
 9974  026210                    .WORD   .S.
                                 MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
 9975            000025          MICPC=MICPC+1
 9976  026210    016777          .WORD   .S.
 9977  026212                    MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
 9978            000026          MICPC=MICPC+1
 9979  026212    016777          .WORD   .S.
 9980  026214                    MOVE    # -1,MEM MARINC
 9981            000027          MICPC=MICPC+1
 9982  026214    016777          .WORD   .S.
 9983  026216                    MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
 9984            000030          MICPC=MICPC+1
 9985  026216    016777          .WORD   .S.
 9986  026220                    MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
 9987            000031          MICPC=MICPC+1
 9988  026220    016525          .WORD   .S.
 9989  026222                    MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
 9990            000032          MICPC=MICPC+1
 9991  026222    016525          .WORD   .S.
 9992  026224                    MOVE    # 125,MEM MARINC
 9993            000033          MICPC=MICPC+1
 9994  026224    016525          .WORD   .S.
 9995  026226                    MOVE    # 125,MEM MARINC        ;RESULT WITH C BIT SET.
 9996            000034          MICPC=MICPC+1
 9997  026226    016525          .WORD   .S.
 9998  026230                    MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
 9999            000035          MICPC=MICPC+1
10000  026230    016652          .WORD   .S.
10001  026232                    MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10002            000036          MICPC=MICPC+1
10003  026232    016525          .WORD   .S.
10004  026234                    MOVE    # -1,MEM MARINC
10005            000037          MICPC=MICPC+1
10006  026234    016777          .WORD   .S.
10007  026236                    MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10008            000040          MICPC=MICPC+1
10009  026236    016777          .WORD   .S.
10010  026240                    MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10011            000041          MICPC=MICPC+1
10012  026240    016525          .WORD   .S,
10013  026242                    MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
10014            000042          MICPC=MICPC+1
10015  026242    016652          .WORD   .S.
10016  026244                    MOVE    # -1,MEM MARINC
10017            000043          MICPC=MICPC+1
10018  026244    016777          .WORD   .S.
10019  026246                    MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10020            000044          MICPC=MICPC+1
10021  026246    016777          .WORD   .S.
10022  026250                    MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10023            000045          MICPC=MICPC+1
10024  026250    016652          .WORD   .S,
10025  026252                    MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10026            000046          MICPC=MICPC+1
10027  026252    016652          .WORD   .S.
```

```
10028  026254                    MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
10029         000047             MICPC=MICPC+1
10030  026254 016652             .WORD   .$.
10031  026256                    MOVE    # 252,MEM MARINC         ;RESULT WITH C BIT SET.
10032         000050             MICPC=MICPC+1
10033  026256 016652             .WORD   .$.
10034  026260                    MOVE    # 7,SPAD <7>     ;SET THE COUNT.
10035         000051             MICPC=MICPC+1
10036  026260 003007             .WORD   .$.
10037
10038  026262                    MOVE    # 0,MLR          ;MAR+0.
10039         000052             MICPC=MICPC+1
10040  026262 010000             .WORD   .$.
10041
10042  026264              2$:   MOVE    # 0,BREG                  ;
10043         000053             MICPC=MICPC+1
10044  026264 000400             .WORD   .$.
10045  026266                    $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
10046         000054             MICPC=MICPC+1
10047  026266 060401             .WORD   .$.!.DD
10048  026270                    MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
10049         000055             MICPC=MICPC+1
10050  026270 057220             .WORD   .$.
10051  026272                    OR      MEM,SPAD <3>,BR.SP,MARINC       ;SPAD  <0>:=$FUNC.
10052         000056             MICPC=MICPC+1
10053  026272 057700             .WORD   .$.
10054  026274                    $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
10055
10056
10057  026274                    SUB2C   SPAD <0>,MEM,NOP
10058         000057             MICPC=MICPC+1
10059  026274 040360             .WORD   .$.
10060  026276                    BZ      3$
10061         000060             MICPC=MICPC+1
10062  026276 101473             .WORD   .$.
10063  026300                    MOVE    MEM,OUT1 <4>    ;GOOD DATA
10064         000061             MICPC=MICPC+1
10065  026300 041224             .WORD   .$.
10066  026302                    MOVE    BREG,OUT1 <5>   ;BAD DATA.
10067         000062             MICPC=MICPC+1
10068  026302 061225             .WORD   .$.
10069  026304                    MOVE    # 15,BREG       ;SET TYPE OF ERROR.
10070         000063             MICPC=MICPC+1
10071  026304 000415             .WORD   .$.
10072  026306                    MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
10073         000064             MICPC=MICPC+1
10074  026306 061223             .WORD   .$.
10075  026310                    MOVE    # 14,BREG       ;LOAD FUNCTION CODE...
10076         000065             MICPC=MICPC+1
10077  026310 000414             .WORD   .$.
10078  026312                    MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
10079         000066             MICPC=MICPC+1
10080  026312 061227             .WORD   .$.
10081  026314                    CALL    ERROR           ;ALU A OR B ERROR...
10082  026314                    MOVE    # <MICPC+3>,BREG
10083         000067             MICPC=MICPC+1
```

```
10084  026314  000471                        .WORD   .$.
10085  026316                                SBR     ERROR
10086          000070                        MICPC=MICPC+1
10087  026316  104400                        .WORD   .$.
10088  026320                                MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
10089          000071                        MICPC=MICPC+1
10090  026320  070204                        .WORD   .$.
10091  026322                                SBR     2$              ;LOOP ON ERROR...
10092          000072                        MICPC=MICPC+1
10093  026322  100453                        .WORD   .$.
10094  026324                          3$:   CALL    SCP1
10095  026324                                MOVE    # <MICPC+3>,BREG
10096          000073                        MICPC=MICPC+1
10097  026324  000475                        .WORD   .$.
10098  026326                                SBR     SCP1
10099          000074                        MICPC=MICPC+1
10100  026326  104427                        .WORD   .$.
10101  026330                                MOVE    SPAD <4>,MLR    ;
10102          000075                        MICPC=MICPC+1
10103  026330  070204                        .WORD   .$.
10104  026332                                SBR     2$              ; SCOPE THE DATA....
10105          000076                        MICPC=MICPC+1
10106  026332  100453                        .WORD   .$.
10107  026334                          6$:   MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
10108          000077                        MICPC=MICPC+1
10109  026334  070204                        .WORD   .$.
10110  026336                                MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
10111          000100                        MICPC=MICPC+1
10112  026336  057220                        .WORD   .$.
10113  026340                                MOVE    # 377,BREG      ;
10114          000101                        MICPC=MICPC+1
10115  026340  000777                        .WORD   .$.
10116  026342                                SPAD    SPAD <2>,BREG   ;SET C BIT...
10117          000102                        MICPC=MICPC+1
10118  026342  060402                        .WORD   .$.!.DD
10119  026344                                OR      MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = A OR B
10120          000103                        MICPC=MICPC+1
10121  026344  057700                        .WORD   .$.
10122  026346                                MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
10123          000104                        MICPC=MICPC+1
10124  026346  057223                        .WORD   .$.
10125  026350                                SIFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
10126
10127
10128  026350                                SUB2C   SPAD <0>,MEM,NOP
10129          000105                        MICPC=MICPC+1
10130  026350  040360                        .WORD   .$.
10131  026352                                BZ      9$
10132          000106                        MICPC=MICPC+1
10133  026352  101521                        .WORD   .$.
10134  026354                                MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
10135          000107                        MICPC=MICPC+1
10136  026354  041224                        .WORD   .$.
10137  026356                                MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
10138          000110                        MICPC=MICPC+1
10139  026356  061225                        .WORD   .$.
```

```
10140 026360                    MOVE    # 23,BREG                ;ERROR TYPE...
10141       000111              MICPC=MICPC+1
10142 026360 000423             .WORD   .$.
10143 026362                    MOVE    BREG,OUT1 <CSR3>        ;
10144       000112              MICPC=MICPC+1
10145 026362 061223             .WORD   .$.
10146 026364                    MOVE    # 14,BREG               ;ALU FUNCTION CODE.
10147       000113              MICPC=MICPC+1
10148 026364 000414             .WORD   .$.
10149 026366                    MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
10150       000114              MICPC=MICPC+1
10151 026366 061227             .WORD   .$.
10152 026370                    CALL    ERROR   ;REPORT ERROR...
10153 026370                    MOVE    # <MICPC+3>,BREG
10154       000115              MICPC=MICPC+1
10155 026370 000517             .WORD   .$.
10156 026372                    $BR     ERROR
10157       000116              MICPC=MICPC+1
10158 026372 104400             .WORD   .$.
10159 026374                    MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
10160       000117              MICPC=MICPC+1
10161 026374 070204             .WORD   .$.
10162 026376                    $BR     6$                      ;LOOP ON ERROR...
10163       000120              MICPC=MICPC+1
10164 026376 100477             .WORD   .$.
10165 026400                9$: CALL    SCP1                    ;SCOPE THE ERROR...
10166 026400                    MOVE    # <MICPC+3>,BREG
10167       000121              MICPC=MICPC+1
10168 026400 000523             .WORD   .$.
10169 026402                    $BR     SCP1
10170       000122              MICPC=MICPC+1
10171 026402 104427             .WORD   .$.
10172 026404                    MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
10173       000123              MICPC=MICPC+1
10174 026404 070204             .WORD   .$.
10175 026406                    $BR     6$              ;SCOPE THE DATA...
10176       000124              MICPC=MICPC+1
10177 026406 100477             .WORD   .$.
10178 026410                    MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
10179       000125              MICPC=MICPC+1
10180 026410 000404             .WORD   .$.
10181 026412                    $ADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
10182 026412 077004             .WORD   .$.!MARINC!.DD
10183 026414                    $DEC    SPAD <7>                ;IS IT DONE??
10184       000126              MICPC=MICPC+1
10185 026414 063167         .WORD   .$.!.DSP
10186 026416                    BZ      4$                      ;YES, SCOPE THE TEST.
10187       000127              MICPC=MICPC+1
10188 026416 101532             .WORD   .$.
10189 026420                    $BR     2$                      ;DO, THE NEXT.
10190       000130              MICPC=MICPC+1
10191 026420 100453             .WORD   .$.
10192 026422                4$: CALL    SCPE                    ;SCOPE THE TEST...
10193 026422                    MOVE    # <MICPC+3>,BREG
10194       000131              MICPC=MICPC+1
10195 026422 000533             .WORD   .$.
```

```
10196  026424                          SBR     SCPE
10197          000132                  MICPC=MICPC+1
10198  026424  104454                  .WORD   .$.
10199  026426                          SBR     1$              ;DO THE NEXT ITERATION...
10200          000133                  MICPC=MICPC+1
10201  026426  100400                  .WORD   .$.
10202  026430                  $ALUT1  0,<A XOR B>,$XOR,0,0,-1,-1,-1,-1,0,0,0,0,-1,-1,-1,-1,0,0,<A XOR B>,1,15
10203  026430                  $XZ
10204
10205
10206                                   ;************************** TEST 47 **************************
10207                                   ;*ALU TEST
10208                                   ;*TEST OF ALU FUNCTION A XOR B WITH C BIT CLEARED.
10209                                   ;*TEST OF ALU FUNCTION A XOR B WITH C BIT SET.
10210                                   ;*ALU FUNCTION (A XOR B)
10211                                   ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
10212                                   ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
10213  026430                  $XZ
10214                                   ;;*********************************************************************
10215
10216  026430                  $TSTN
10217                                   ;  TEST 47
10218                                   ;--------------
10219  026430  012737  000047  001202  TST47:  MOV     #47,$TSTNM              ; LOAD THE NO. OF THIS TEST
10220  026436  012737  026756  001442          MOV     #TST50,NEXT             ; POINT TO THE START OF NEXT TEST.
10221                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
10222  026444  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
10223  026450  026464                          MCT47
10224  026452  104022                          ERROR   22                      ;TIME OUT ERROR...
10225  026454  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
10226  026460  000177  152756                  JMP     @NEXT                   ;GO TO NEXT TEST...
10227  026464                          MCT47:
10228  026464                          1$:     MOVE    # 0,MLR          ;SET MAR<LO.
10229          000000                  MICPC=MICPC+1
10230  026464  010000                  .WORD   .$.
10231  026466                          MOVE    # 0,MPR          ;SET MAR<HI.
10232          000001                  MICPC=MICPC+1
10233  026466  004000                  .WORD   .$.
10234  026470                          MOVE    # 0,BREG                 ;
10235          000002                  MICPC=MICPC+1
10236  026470  000400                  .WORD   .$.
10237  026472                          MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
10238          000003                  MICPC=MICPC+1
10239  026472  063236                  .WORD   .$.
10240  026474                          MOVE    BREG,SPAD <0>   ;
10241          000004                  MICPC=MICPC+1
10242  026474  063220                  .WORD   .$.
10243  026476                          MOVE    BREG,SPAD <1>   ;
10244          000005                  MICPC=MICPC+1
10245  026476  063221                  .WORD   .$.
10246  026500                          MOVE    BREG,SPAD <2>   ;
10247          000006                  MICPC=MICPC+1
10248  026500  063222                  .WORD   .$.
10249  026502                          SDEC    SPAD <2>                ;
10250          000007                  MICPC=MICPC+1
10251  026502  063162          .WORD   .$.!.DSP
```

```
10252 026504                    MOVE    BREG,SPAD (4)           ;
10253        000010             MICPC=MICPC+1
10254 026504 063224             .WORD   .S.
10255 026506                    MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10256        000011             MICPC=MICPC+1
10257 026506 016400             .WORD   .S.
10258 026510                    MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
10259        000012             MICPC=MICPC+1
10260 026510 016400             .WORD   .S.
10261 026512                    MOVE    # 0,MEM MARINC
10262        000013             MICPC=MICPC+1
10263 026512 016400             .WORD   .S.
10264 026514                    MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
10265        000014             MICPC=MICPC+1
10266 026514 016400             .WORD   .S.
10267 026516                    MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
10268        000015             MICPC=MICPC+1
10269 026516 016777             .WORD   .S.
10270 026520                    MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10271        000016             MICPC=MICPC+1
10272 026520 016400             .WORD   .S.
10273 026522                    MOVE    # -1,MEM MARINC
10274        000017             MICPC=MICPC+1
10275 026522 016777             .WORD   .S.
10276 026524                    MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10277        000020             MICPC=MICPC+1
10278 026524 016777             .WORD   .S.
10279 026526                    MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10280        000021             MICPC=MICPC+1
10281 026526 016400             .WORD   .S.
10282 026530                    MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
10283        000022             MICPC=MICPC+1
10284 026530 016777             .WORD   .S.
10285 026532                    MOVE    # -1,MEM MARINC
10286        000023             MICPC=MICPC+1
10287 026532 016777             .WORD   .S.
10288 026534                    MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10289        000024             MICPC=MICPC+1
10290 026534 016777             .WORD   .S.
10291 026536                    MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
10292        000025             MICPC=MICPC+1
10293 026536 016777             .WORD   .S.
10294 026540                    MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
10295        000026             MICPC=MICPC+1
10296 026540 016777             .WORD   .S.
10297 026542                    MOVE    # 0,MEM MARINC
10298        000027             MICPC=MICPC+1
10299 026542 016400             .WORD   .S.
10300 026544                    MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
10301        000030             MICPC=MICPC+1
10302 026544 016400             .WORD   .S.
10303 026546                    MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
10304        000031             MICPC=MICPC+1
10305 026546 016525             .WORD   .S.
10306 026550                    MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
10307        000032             MICPC=MICPC+1
```

```
10308  026550  016525                  .WORD   .S.
10309  026552                          MOVE    # 0,MEM MARINC
10310          000033                  MICPC=MICPC+1
10311  026552  016400                  .WORD   .S.
10312  026554                          MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
10313          000034                  MICPC=MICPC+1
10314  026554  016400                  .WORD   .S.
10315  026556                          MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10316          000035                  MICPC=MICPC+1
10317  026556  016652                  .WORD   .S.
10318  026560                          MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10319          000036                  MICPC=MICPC+1
10320  026560  016525                  .WORD   .S.
10321  026562                          MOVE    # -1,MEM MARINC
10322          000037                  MICPC=MICPC+1
10323  026562  016777                  .WORD   .S.
10324  026564                          MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10325          000040                  MICPC=MICPC+1
10326  026564  016777                  .WORD   .S.
10327  026566                          MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10328          000041                  MICPC=MICPC+1
10329  026566  016525                  .WORD   .S.
10330  026570                          MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
10331          000042                  MICPC=MICPC+1
10332  026570  016652                  .WORD   .S.
10333  026572                          MOVE    # -1,MEM MARINC
10334          000043                  MICPC=MICPC+1
10335  026572  016777                  .WORD   .S.
10336  026574                          MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10337          000044                  MICPC=MICPC+1
10338  026574  016777                  .WORD   .S.
10339  026576                          MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10340          000045                  MICPC=MICPC+1
10341  026576  016652                  .WORD   .S.
10342  026600                          MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10343          000046                  MICPC=MICPC+1
10344  026600  016652                  .WORD   .S.
10345  026602                          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10346          000047                  MICPC=MICPC+1
10347  026602  016400                  .WORD   .S.
10348  026604                          MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
10349          000050                  MICPC=MICPC+1
10350  026604  016400                  .WORD   .S.
10351  026606                          MOVE    # 7,SPAD <7>    ;SET THE COUNT.
10352          000051                  MICPC=MICPC+1
10353  026606  003007                  .WORD   .S.
10354
10355  026610                          MOVE    # 0,MLR         ;MAR←0.
10356          000052                  MICPC=MICPC+1
10357  026610  010000                  .WORD   .S.
10358
10359  026612                  2$:     MOVE    # 0,BREG                 ;
10360          000053                  MICPC=MICPC+1
10361  026612  000400                  .WORD   .S.
10362  026614                          $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
10363          000054                  MICPC=MICPC+1
```

```
10364  026614  060401                  .WORD   .$.!.DD
10365  026616                          MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
10366          000055                  MICPC=MICPC+1
10367  026616  057220                  .WORD   .$.
10370  026620                          $XOR    MEM,SPAD <0>,BR.SP,MARINC       ;SPAD  <0>:=$FUNC.
10371          000056                  MICPC=MICPC+1
10372  026620  057720                  .WORD   .$.
10373  026622                          $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
10374
10375  026622                          SUB2C   SPAD <0>,MEM,NOP
10376          000057                  MICPC=MICPC+1
10377  026622  040360                  .WORD   .$.
10400  026624                          BZ      3$
10401          000060                  MICPC=MICPC+1
10402  026624  101473                  .WORD   .$.
10403  026626                          MOVE    MEM,OUT1 <4>    ;GOOD DATA
10404          000061                  MICPC=MICPC+1
10405  026626  041224                  .WORD   .$.
10406  026630                          MOVE    BREG,OUT1 <5>   ;BAD DATA.
10407          000062                  MICPC=MICPC+1
10410  026630  061225                  .WORD   .$.
10411  026632                          MOVE    # 15,BREG       ;SET TYPE OF ERROR.
10412          000063                  MICPC=MICPC+1
10413  026632  000415                  .WORD   .$.
10414  026634                          MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
10415          000064                  MICPC=MICPC+1
10416  026634  061223                  .WORD   .$.
10417  026636                          MOVE    # 15,BREG       ;LOAD FUNCTION CODE...
10420          000065                  MICPC=MICPC+1
10421  026636  000415                  .WORD   .$.
10422  026640                          MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
10423          000066                  MICPC=MICPC+1
10424  026640  061227                  .WORD   .$.
10425  026642                          CALL    EROR            ;ALU A XOR B ERROR...
10426  026642                          MOVE    # <MICPC+3>,BREG
10427          000067                  MICPC=MICPC+1
10430  026642  000471                  .WORD   .$.
10431  026644                          $BR     EROR
10432          000070                  MICPC=MICPC+1
10433  026644  104400                  .WORD   .$.
10434  026646                          MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
10435          000071                  MICPC=MICPC+1
10436  026646  070204                  .WORD   .$.
10437  026650                          $BR     2$              ;LOOP ON ERROR...
10440          000072                  MICPC=MICPC+1
10441  026650  100453                  .WORD   .$.
10442  026652                  3$:     CALL    SCP1
10443  026652                          MOVE    # <MICPC+3>,BREG
10444          000073                  MICPC=MICPC+1
10445  026652  000475                  .WORD   .$.
10446  026654                          $BR     SCP1
10447          000074                  MICPC=MICPC+1
10450  026654  104427                  .WORD   .$.
10451  026656                          MOVE    SPAD <4>,MLR    ;
10452          000075                  MICPC=MICPC+1
```

```
10420  026656  070204                    .WORD   .$.
10421  026660                            $BR     2$                  ; SCOPE THE DATA....
10422          000076                    MICPC=MICPC+1
10423  026660  100453                    .WORD   .$.
10424  026662                    6$:     MOVE    SPAD <4>,MLR        ;RESET DATA POINTER...
10425          000077                    MICPC=MICPC+1
10426  026662  070204                    .WORD   .$.
10427  026664                            MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
10428          000100                    MICPC=MICPC+1
10429  026664  057220                    .WORD   .$.
10430  026666                            MOVE    # 377,BREG          ;
10431          000101                    MICPC=MICPC+1
10432  026666  000777                    .WORD   .$.
10433  026670                            $AOD    SPAD <2>,BREG       ;SET C BIT...
10434          000102                    MICPC=MICPC+1
10435  026670  060402                    .WORD   .$.!.DO
10436  026672                            $XOR    MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = A XOR B
10437          000103                    MICPC=MICPC+1
10438  026672  057720                    .WORD   .$.
10439  026674                            MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
10440          000104                    MICPC=MICPC+1
10441  026674  057223                    .WORD   .$.
10442  026676                            $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
10443
10444
10445  026676                            SUB2C   SPAD <0>,MEM,NOP
10446          000105                    MICPC=MICPC+1
10447  026676  040360                    .WORD   .$.
10448  026700                            BZ      9$
10449          000106                    MICPC=MICPC+1
10450  026700  101521                    .WORD   .$.
10451  026702                            MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
10452          000107                    MICPC=MICPC+1
10453  026702  041224                    .WORD   .$.
10454  026704                            MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
10455          000110                    MICPC=MICPC+1
10456  026704  061225                    .WORD   .$.
10457  026706                            MOVE    # 23,BREG               ;ERROR TYPE...
10458          000111                    MICPC=MICPC+1
10459  026706  000423                    .WORD   .$.
10460  026710                            MOVE    BREG,OUT1 <CSR3>        ;
10461          000112                    MICPC=MICPC+1
10462  026710  061223                    .WORD   .$.
10463  026712                            MOVE    # 15,BREG               ;ALU FUNCTION CODE.
10464          000113                    MICPC=MICPC+1
10465  026712  000415                    .WORD   .$.
10466  026714                            MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
10467          000114                    MICPC=MICPC+1
10468  026714  061227                    .WORD   .$.
10469  026716                            CALL    EROR    ;REPORT ERROR...
10470  026716                            MOVE    # <MICPC+3>,BREG
10471          000115                    MICPC=MICPC+1
10472  026716  000517                    .WORD   .$.
10473  026720                            SBR     EROR
10474          000116                    MICPC=MICPC+1
10475  026720  104400                    .WORD   .$.
```

```
10476 026722                    MOVE    SPAD (4),MLR            ;RESTORE DATA POINTER.
10477         000117            MICPC=MICPC+1
10478 026722  070204            .WORD   .$.
10479 026724                    SBR     6$                      ;LOOP ON ERROR...
10480         000120            MICPC=MICPC+1
10481 026724  100477            .WORD   $.
10482 026726              9$:   CALL    $CP1              ;SCOPE THE ERROR...
10483 026726                    MOVE    # (MICPC+3),BREG
10484         000121            MICPC=MICPC+1
10485 026726  000523            .WORD   $.
10486 026730                    SBR     $CP1
10487         000122            MICPC=MICPC+1
10488 026730  104427            .WORD   .$.
10489 026732                    MOVE    SPAD (4),MLR    ;RESTORE DATA POINTER...
10490         000123            MICPC=MICPC+1
10491 026732  070204            .WORD   .$.
10492 026734                    SBR     6$                      ;SCOPE THE DATA...
10493         000124            MICPC=MICPC+1
10494 026734  100477            .WORD   .$.
10495 026736                    MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
10496         000125            MICPC=MICPC+1
10497 026736  000404            .WORD   .$.
10498 026740                    SADD    BREG,SPAD (4),MARINC    ;ALSO DATA POINTER.
10499 026740  077004            .WORD   .$.!MARINC!.DD
10500 026742                    SDEC    SPAD (7)                ;IS IT DONE??
10501         000126            MICPC=MICPC+1
10502 026742  063167      .WORD .$.!.DSP
10503 026744                    BZ      4$                      ;YES, SCOPE THE TEST.
10504         000127            MICPC=MICPC+1
10505 026744  101532            .WORD   $.
10506 026746                    SBR     2$                      ;DO, THE NEXT.
10507         000130            MICPC=MICPC+1
10508 026746  100453            .WORD   $.
10509 026750              4$:   CALL    SCPE              ;SCOPE THE TEST...
10510 026750                    MOVE    @ (MICPC+3),BREG
10511         000131            MICPC=MICPC+1
10512 026750  000533            .WORD   $.
10513 026752                    SBR     SCPE
10514         000132            MICPC=MICPC+1
10515 026752  104454            .WORD   .$.
10516 026754                    SBR     1$                      ;DO THE NEXT ITERATION...
10517         000133            MICPC=MICPC+1
10518 026754  100400            .WORD   $.
10519 026756              SALUT1: 0,<ADD>,$ADD,0,0,-1,-1,-1,-1,376,376,252,252,-1,-1,-1,-1,124,124,<A PLUS B>,1,00
10520 026756              SXZ
10521
10522
10523                           ;************************** TEST 50 **************************
10524                           ;*ALU TEST
10525                           ;*TEST OF ALU FUNCTION ADD WITH C BIT CLEARED.
10526                           ;*TEST OF ALU FUNCTION ADD WITH C BIT SET.
10527                           ;*ALU FUNCTION (A PLUS B)
10528                           ;*LOAD MAIN MEMORY 16 WORDS OF DATA,
10529                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
10530 026756              SXZ
10531                           ;;*************************************************************
```

```
10532
10533  026756                                   $TSTN
10534
10535                                           ;  TEST 50
10536                                           ;--------------
10537  026756  012737  000050  001202  TST50:  MOV     #50,$TSTNM              ; LOAD THE NO. OF THIS TEST
10540  026764  012737  027304  001442          MOV     #TST51,NEXT             ; POINT TO THE START OF NEXT TEST.
10541                                                                   ;R1 CONTAINS BASE KMC11 ADDRESS
10542  026772  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
10543  026776  027012                          MCT50
10544  027000  104022                          ERROR   22                      ;TIME OUT ERROR...
10545  027002  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
10546  027006  000177  152430                  JMP     @NEXT                   ;GO TO NEXT TEST...
10547  027012                          MCT50:
10550  027012                          1$:     MOVE    # 0,MLR          ;SET MAR<-LO.
10551  027012  000000                          MICPC=MICPC+1
10552  027014  010000                          .WORD   .$.
10553  027014                                  MOVE    # 0,MPR          ;SET MAR<-HI.
10554  027016  000001                          MICPC=MICPC+1
10555  027016  004000                          .WORD   .$.
10556  027020                                  MOVE    # 0,BREG                 ;
10557  027020  000002                          MICPC=MICPC+1
10560  027020  000400                          .WORD   .$.
10561  027022                                  MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
10562  027022  000003                          MICPC=MICPC+1
10563  027022  063236                          .WORD   .$.
10564  027024                                  MOVE    BREG,SPAD <0>   ;
10565  027024  000004                          MICPC=MICPC+1
10566  027024  063220                          .WORD   .$.
10567  027026                                  MOVE    BREG,SPAD <1>   ;
10570  027026  000005                          MICPC=MICPC+1
10571  027026  063221                          .WORD   .$.
10572  027030                                  MOVE    BREG,SPAD <2>   ;
10573  027030  000006                          MICPC=MICPC+1
10574  027030  063222                          .WORD   .$.
10575  027032                                  SDEC    SPAD <2>                ;
10576  027032  000007                          MICPC=MICPC+1
10577  027032  063162          .WORD           .$.!.DSP
10600  027034                                  MOVE    BREG,SPAD <4>           ;
10601  027034  000010                          MICPC=MICPC+1
10602  027034  063224                          .WORD   .$.
10603  027036                                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10604  027036  000011                          MICPC=MICPC+1
10605  027036  016400                          .WORD   .$.
10606  027040                                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
10607  027040  000012                          MICPC=MICPC+1
10610  027040  016400                          .WORD   .$.
10611  027042                                  MOVE    # 0,MEM MARINC
10612  027042  000013                          MICPC=MICPC+1
10613  027042  016400                          .WORD   .$.
10614  027044                                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
10615  027044  000014                          MICPC=MICPC+1
10616  027044  016400                          .WORD   .$.
10617  027046                                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
10620  027046  000015                          MICPC=MICPC+1
10621  027046  016777                          .WORD   .$.
10622  027050                                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
```

```
10588                   000016          MICPC=MICPC+1
10589   027046  016400                  .WORD   .S.
10590   027050                          MOVE    # -1,MEM MARINC
10591                   000017          MICPC=MICPC+1
10592   027050  016777                  .WORD   .S.
10593   027052                          MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10594                   000020          MICPC=MICPC+1
10595   027052  016777                  .WORD   .S.
10596   027054                          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10597                   000021          MICPC=MICPC+1
10598   027054  016400                  .WORD   .S.
10599   027056                          MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
10600                   000022          MICPC=MICPC+1
10601   027056  016777                  .WORD   .S.
10602   027060                          MOVE    # -1,MEM MARINC
10603                   000023          MICPC=MICPC+1
10604   027060  016777                  .WORD   .S.
10605   027062                          MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10606                   000024          MICPC=MICPC+1
10607   027062  016777                  .WORD   .S.
10608   027064                          MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
10609                   000025          MICPC=MICPC+1
10610   027064  016777                  .WORD   .S.
10611   027066                          MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
10612                   000026          MICPC=MICPC+1
10613   027066  016777                  .WORD   .S.
10614   027070                          MOVE    # 376,MEM MARINC
10615                   000027          MICPC=MICPC+1
10616   027070  016776                  .WORD   .S.
10617   027072                          MOVE    # 376,MEM MARINC         ;RESULT WITH C BIT SET.
10618                   000030          MICPC=MICPC+1
10619   027072  016776                  .WORD   .S.
10620   027074                          MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
10621                   000031          MICPC=MICPC+1
10622   027074  016525                  .WORD   .S.
10623   027076                          MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
10624                   000032          MICPC=MICPC+1
10625   027076  016525                  .WORD   .S.
10626   027100                          MOVE    # 252,MEM MARINC
10627                   000033          MICPC=MICPC+1
10628   027100  016652                  .WORD   .S.
10629   027102                          MOVE    # 252,MEM MARINC         ;RESULT WITH C BIT SET.
10630                   000034          MICPC=MICPC+1
10631   027102  016652                  .WORD   .S.
10632   027104                          MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
10633                   000035          MICPC=MICPC+1
10634   027104  016652                  .WORD   .S.
10635   027106                          MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
10636                   000036          MICPC=MICPC+1
10637   027106  016525                  .WORD   .S.
10638   027110                          MOVE    # -1,MEM MARINC
10639                   000037          MICPC=MICPC+1
10640   027110  016777                  .WORD   .S.
10641   027112                          MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10642                   000040          MICPC=MICPC+1
10643   027112  016777                  .WORD   .S.
```

```
10644 027114
10645         000041
10646 027114  016525          MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
                              MICPC=MICPC+1
                              .WORD   .S.
10647 027116
10650         000042          MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY
10651 027116  016652          MICPC=MICPC+1
10652 027120                  .WORD   .S.
10653         000043          MOVE    # -1,MEM MARINC
10654 027120  016777          MICPC=MICPC+1
10655 027122                  .WORD   .S.
10656         000044          MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10657 027122  016777          MICPC=MICPC+1
10660 027124                  .WORD   .S.
10661         000045          MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
10662 027124  016652          MICPC=MICPC+1
10663 027126                  .WORD   .S.
10664         000046          MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
10665 027126  016652          MICPC=MICPC+1
10666 027130                  .WORD   .S.
10667         000047          MOVE    # 124,MEM MARINC         ;LOAD THE DATA IN MEMORY.
10670 027130  016524          MICPC=MICPC+1
10671 027132                  .WORD   .S.
10672         000050          MOVE    # 124,MEM MARINC         ;RESULT WITH C BIT SET.
10673 027132  016524          MICPC=MICPC+1
10674 027134                  .WORD   .S.
10675         000051          MOVE    # 7,SPAD <7>     ;SET THE COUNT.
10676 027134  003007          MICPC=MICPC+1
10677                         .WORD   .S.
10700 027136
10701         000052          MOVE    # 0,MLR          ;MAR+0.
10702 027136  010000          MICPC=MICPC+1
10703                         .WORD   .S.
10704 027140
10705         000053    2$:   MOVE    # 0,BREG                  ;
10706 027140  000400          MICPC=MICPC+1
10707 027142                  .WORD   .S.
10710         000054          $AOD    SPAD <1>,BREG   ;CLEAR C BIT.
10711 027142  060401          MICPC=MICPC+1
10712 027144                  .WORD   .S.!.DD
10713         000055          MOVE    MEM,SPAD <0> MARINC      ;GET THE FIRST OPERAND.
10714 027144  057220          MICPC=MICPC+1
10715 027146                  .WORD   .S.
10716         000056          $AOD    MEM,SPAD <0>,BR.SP,MARINC        ;SPAD  <0>:=$FUNC.
10717 027146  057400          MICPC=MICPC+1
10720 027150                  .WORD   .S.
10721                         $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
10722
10723
10724 027150
10725         000057          SUB2C   SPAD <0>,MEM,NOP
10726 027150  040360          MICPC=MICPC+1
10727 027152                  .WORD   .S.
10730         000060          BZ      3$
10731 027152  101473          MICPC=MICPC+1
10732 027154                  .WORD   .S.
10733         000061          MOVE    MEM,OUT1 <4>     ;GOOD DATA
10734 027154  041224          MICPC=MICPC+1
                              .WORD   .S.
```

```
10700  027156
10701          000062
10702  027156  061225
10703  027160
10704          000063
10705  027160  000415
10706  027162
10707          000064
10708  027162  061223
10709  027164
10710          000065
10711  027164  000400
10712  027166
10713          000066
10714  027166  061227
10715  027170
10716  027170
10717          000067
10718  027170  000471
10719  027172
10720          000070
10721  027172  104400
10722  027174
10723          000071
10724  027174  070204
10725  027176
10726          000072
10727  027176  100453
10728  027200
10729  027200
10730          000073
10731  027200  000475
10732  027202
10733          000074
10734  027202  104427
10735  027204
10736          000075
10737  027204  070204
10738  027206
10739          000076
10740  027206  100453
10741  027210
10742          000077
10743  027210  070204
10744  027212
10745          000100
10746  027212  057220
10747  027214
10748          000101
10749  027214  000777
10750  027216
10751          000102
10752  027216  060402
10753  027220
10754          000103
10755  027220  057400
```

```
        MOVE    BREG,OUT1 <5>   ;BAD DATA.
        MICPC=MICPC+1
        .WORD   .S.
        MOVE    # 15,BREG       ;SET TYPE OF ERROR.
        MICPC=MICPC+1
        .WORD   .S.
        MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
        MICPC=MICPC+1
        .WORD   .S.
        MOVE    # 00,BREG       ;LOAD FUNCTION CODE...
        MICPC=MICPC+1
        .WORD   .S.
        MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
        MICPC=MICPC+1
        .WORD   .S.
        CALL    ERROR           ;ALU ADD ERROR...
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .S.
        SBR     ERROR
        MICPC=MICPC+1
        .WORD   .S.
        MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
        MICPC=MICPC+1
        .WORD   .S.
        SBR     25              ;LOOP ON ERROR...
        MICPC=MICPC+1
        .WORD   .S.
35:     CALL    SCP1
        MOVE    # <MICPC+3>,BREG
        MICPC=MICPC+1
        .WORD   .S.
        SBR     SCP1
        MICPC=MICPC+1
        .WORD   .S.
        MOVE    SPAD <4>,MLR    ;
        MICPC=MICPC+1
        .WORD   .S.
        SBR     25              ; SCOPE THE DATA....
        MICPC=MICPC+1
        .WORD   .S.
65:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
        MICPC=MICPC+1
        .WORD   .S.
        MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
        MICPC=MICPC+1
        .WORD   .S.
        MOVE    # 377,BREG      ;
        MICPC=MICPC+1
        .WORD   .S.
        SADD    SPAD <2>,BREG   ;SET C BIT...
        MICPC=MICPC+1
        .WORD   .S.!.DD
        SADD    MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = ADD
        MICPC=MICPC+1
        .WORD   .S.
```

```
10756  027222                  MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
10757          000104           MICPC=MICPC+1
10758  027222  057223           .WORD   .$.
10759  027224                  $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
10760
10761
10762  027224                  SUB2C   SPAD <0>,MEM,NOP
10763          000105           MICPC=MICPC+1
10764  027224  040360           .WORD   .$.
10765  027226                  BZ      9$
10766          000106           MICPC=MICPC+1
10767  027226  101521           .WORD   .$.
10768  027230                  MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
10769          000107           MICPC=MICPC+1
10770  027230  041224           .WORD   .$.
10771  027232                  MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
10772          000110           MICPC=MICPC+1
10773  027232  061225           .WORD   .$.
10774  027234                  MOVE    # 23,BREG               ;ERROR TYPE...
10775          000111           MICPC=MICPC+1
10776  027234  000423           .WORD   .$.
10777  027236                  MOVE    BREG,OUT1 <CSR3>        ;
10778          000112           MICPC=MICPC+1
10779  027236  061223           .WORD   .$.
10780  027240                  MOVE    # DD,BREG               ;ALU FUNCTION CODE.
10781          000113           MICPC=MICPC+1
10782  027240  000400           .WORD   .$.
10783  027242                  MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
10784          000114           MICPC=MICPC+1
10785  027242  061227           .WORD   .$.
10786  027244                  CALL    ERROR   ;REPORT ERROR...
10787  027244                  MOVE    # <MICPC+3>,BREG
10788          000115           MICPC=MICPC+1
10789  027244  000517           .WORD   .$.
10790  027246                  SBR     ERROR
10791          000116           MICPC=MICPC+1
10792  027246  104400           .WORD   .$.
10793  027250                  MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
10794          000117           MICPC=MICPC+1
10795  027250  070204           .WORD   .$.
10796  027252                  SBR     6$                      ;LOOP ON ERROR...
10797          000120           MICPC=MICPC+1
10798  027252  100477           .WORD   .$.
10799  027254          9$:     CALL    SCP1            ;SCOPE THE ERROR...
10800  027254                  MOVE    # <MICPC+3>,BREG
10801          000121           MICPC=MICPC+1
10802  027254  000523           .WORD   .$.
10803  027256                  SBR     SCP1
10804          000122           MICPC=MICPC+1
10805  027256  104427           .WORD   .$.
10806  027260                  MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
10807          000123           MICPC=MICPC+1
10808  027260  070204           .WORD   .$.
10809  027262                  SBR     6$              ;SCOPE THE DATA...
10810          000124           MICPC=MICPC+1
10811  027262  100477           .WORD   .$.
```

```
10812  027264                                  MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
10813         000125                           MICPC=MICPC+1
10814  027264 000404                           .WORD   .$.
10815  027266                                  SADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
10816  027266 077004                           .WORD   .$.!MARINC!.DO
10817  027270                                  SDEC    SPAD <7>                ;IS IT DONE??
10818         000126                           MICPC=MICPC+1
10819  027270 063167                    .WORD  .$.!.DSP
10820  027272                                  BZ      4$                      ;YES, SCOPE THE TEST.
10821         000127                           MICPC=MICPC+1
10822  027272 101532                           .WORD   .$.
10823  027274                                  SBR     2$                      ;DO, THE NEXT.
10824         000130                           MICPC=MICPC+1
10825  027274 100453                           .WORD   .$.
10826  027276                           4$:    CALL    SCPE                    ;SCOPE THE TEST...
10827  027276                                  MOVE    # <MICPC+3>,BREG
10828         000131                           MICPC=MICPC+1
10829  027276 000533                           .WORD   .$.
10830  027300                                  SBR     SCPE
10831         000132                           MICPC=MICPC+1
10832  027300 104454                           .WORD   .$.
10833  027302                                  SBR     1$                      ;DO THE NEXT ITERATION...
10834         000133                           MICPC=MICPC+1
10835  027302 100400                           .WORD   .$.
10836  027304                           SALUT1 0,<2A W/C>,$ROL,0,1,376,-1,0,1,376,-1,252,253,124,125,252,253,124,125,<A PLUS A
10837  027304                           $XZ
10838
10839
10840                                           ;************************** TEST 51 **************************
10841                                           ;*ALU TEST
10842                                           ;*TEST OF ALU FUNCTION 2A W/C WITH C BIT CLEARED.
10843                                           ;*TEST OF ALU FUNCTION 2A W/C WITH C BIT SET.
10844                                           ;*ALU FUNCTION (A PLUS A PLUS C)
10845                                           ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
10846                                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
10847  027304                           $XZ
10848                                           ;;*********************************************************************
10849
10850  027304                           $TSTN
10851
10852                                           ;  TEST 51
10853                                           ;-------------
10854  027304 012737  000051  001202    TST51: MOV     #51,$TSTNM              ; LOAD THE NO. OF THIS TEST
10855  027312 012737  027632  001442           MOV     #TST52,NEXT             ; POINT TO THE START OF NEXT TEST.
10856                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
10857  027320 004737  035536                   JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
10858  027324 027340                           MCT51
10859  027326 104022                           ERROR   22                      ;TIME OUT ERROR...
10860  027330 012706  001200                   MOV     #STACK,SP               ;RESET STACK...
10861  027334 000177  152102                   JMP     @NEXT                   ;GO TO NEXT TEST...
10862  027340                           MCT51:
10863         000000                    1$:    MOVE    # 0,MLR          ;SET MAR<LO.
10864  027340 010000                           MICPC=MICPC+1
10865  027342                                  .WORD   .$.
10866         000001                           MOVE    # 0,MPR          ;SET MAR<HI.
10867  027342 004000                           MICPC=MICPC+1
                                               .WORD   .$.
```

```
10868 027344                    MOVE    # 0,BREG                ;
10869         000002            MICPC=MICPC+1
10870 027344  000400            .WORD   .$.
10871 027346                    MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
10872         000003            MICPC=MICPC+1
10873 027346  063236            .WORD   .$.
10874 027350                    MOVE    BREG,SPAD <0>   ;
10875         000004            MICPC=MICPC+1
10876 027350  063220            .WORD   .$.
10877 027352                    MOVE    BREG,SPAD <1>   ;
10878         000005            MICPC=MICPC+1
10879 027352  063221            .WORD   .$.
10880 027354                    MOVE    BREG,SPAD <2>   ;
10881         000006            MICPC=MICPC+1
10882 027354  063222            .WORD   .$.
10883 027356                    SDEC    SPAD <2>                ;
10884         000007            MICPC=MICPC+1
10885 027356  063162      .WORD .$.!.DSP
10886 027360                    MOVE    BREG,SPAD <4>           ;
10887         000010            MICPC=MICPC+1
10888 027360  063224            .WORD   .$.
10889 027362                    MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10890         000011            MICPC=MICPC+1
10891 027362  016400            .WORD   .$.
10892 027364                    MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
10893         000012            MICPC=MICPC+1
10894 027364  016400            .WORD   .$.
10895 027366                    MOVE    # 0,MEM MARINC
10896         000013            MICPC=MICPC+1
10897 027366  016400            .WORD   .$.
10898 027370                    MOVE    # 1,MEM MARINC  ;RESULT WITH C BIT SET.
10899         000014            MICPC=MICPC+1
10900 027370  016401            .WORD   .$.
10901 027372                    MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
10902         000015            MICPC=MICPC+1
10903 027372  016777            .WORD   .$.
10904 027374                    MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10905         000016            MICPC=MICPC+1
10906 027374  016400            .WORD   .$.
10907 027376                    MOVE    # 376,MEM MARINC
10908         000017            MICPC=MICPC+1
10909 027376  016776            .WORD   .$.
10910 027400                    MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10911         000020            MICPC=MICPC+1
10912 027400  016777            .WORD   .$.
10913 027402                    MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
10914         000021            MICPC=MICPC+1
10915 027402  016400            .WORD   .$.
10916 027404                    MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
10917         000022            MICPC=MICPC+1
10918 027404  016777            .WORD   .$.
10919 027406                    MOVE    # 0,MEM MARINC
10920         000023            MICPC=MICPC+1
10921 027406  016400            .WORD   .$.
10922 027410                    MOVE    # 1,MEM MARINC  ;RESULT WITH C BIT SET.
10923         000024            MICPC=MICPC+1
```

```
10924  027410  016401                .WORD   .$.
10925  027412                        MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
10926          000025                MICPC=MICPC+1
10927  027412  016777                .WORD   .$.
10928  027414                        MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
10929          000026                MICPC=MICPC+1
10930  027414  016777                .WORD   .$.
10931  027416                        MOVE    # 376,MEM MARINC
10932          000027                MICPC=MICPC+1
10933  027416  016776                .WORD   .$.
10934  027420                        MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
10935          000030                MICPC=MICPC+1
10936  027420  016777                .WORD   .$.
10937  027422                        MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10938          000031                MICPC=MICPC+1
10939  027422  016525                .WORD   .$.
10940  027424                        MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10941          000032                MICPC=MICPC+1
10942  027424  016525                .WORD   .$.
10943  027426                        MOVE    # 252,MEM MARINC
10944          000033                MICPC=MICPC+1
10945  027426  016652                .WORD   .$.
10946  027430                        MOVE    # 253,MEM MARINC        ;RESULT WITH C BIT SET.
10947          000034                MICPC=MICPC+1
10948  027430  016653                .WORD   .$.
10949  027432                        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10950          000035                MICPC=MICPC+1
10951  027432  016652                .WORD   .$.
10952  027434                        MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10953          000036                MICPC=MICPC+1
10954  027434  016525                .WORD   .$.
10955  027436                        MOVE    # 124,MEM MARINC
10956          000037                MICPC=MICPC+1
10957  027436  016524                .WORD   .$.
10958  027440                        MOVE    # 125,MEM MARINC        ;RESULT WITH C BIT SET.
10959          000040                MICPC=MICPC+1
10960  027440  016525                .WORD   .$.
10961  027442                        MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10962          000041                MICPC=MICPC+1
10963  027442  016525                .WORD   .$.
10964  027444                        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
10965          000042                MICPC=MICPC+1
10966  027444  016652                .WORD   .$.
10967  027446                        MOVE    # 252,MEM MARINC
10968          000043                MICPC=MICPC+1
10969  027446  016652                .WORD   .$.
10970  027450                        MOVE    # 253,MEM MARINC        ;RESULT WITH C BIT SET.
10971          000044                MICPC=MICPC+1
10972  027450  016653                .WORD   .$.
10973  027452                        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10974          000045                MICPC=MICPC+1
10975  027452  016652                .WORD   .$.
10976  027454                        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
10977          000046                MICPC=MICPC+1
10978  027454  016652                .WORD   .$.
10979  027456                        MOVE    # 124,MEM MARINC        ;LOAD THE DATA IN MEMORY.
```

```
10980           000047                  MICPC=MICPC+1
10981  027456   016524                  .WORD   .$.
10982  027460                           MOVE    # 125,MEM MARINC        ;RESULT WITH C BIT SET.
10983           000050                  MICPC=MICPC+1
10984  027460   016525                  .WORD   .$.
10985  027462                           MOVE    # 7,SPAD <7>    ;SET THE COUNT.
10986           000051                  MICPC=MICPC+1
10987  027462   003007                  .WORD   .$.
10988
10989  027464                           MOVE    # 0,MLR         ;MAR←0.
10990           000052                  MICPC=MICPC+1
10991  027464   010000                  .WORD   .$.
10992
10993  027466                    2$:    MOVE    # 0,BREG                 ;
10994           000053                  MICPC=MICPC+1
10995  027466   000400                  .WORD   .$.
10996  027470                           $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
10997           000054                  MICPC=MICPC+1
10998  027470   060401                  .WORD   .$.!.DD
10999  027472                           MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
11000           000055                  MICPC=MICPC+1
11001  027472   057220                  .WORD   .$.
11002  027474                           $ROL    SPAD <0>,BR.SP,MARINC   ;
11003           000056                  MICPC=MICPC+1
11004  027474   077540                  .WORD   .$.
11005  027476                           $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
11006
11007
11008  027476                           SUB2C   SPAD <0>,MEM,NOP
11009           000057                  MICPC=MICPC+1
11010  027476   040360                  .WORD   .$.
11011  027500                           BZ      3$
11012           000060                  MICPC=MICPC+1
11013  027500   101473                  .WORD   .$.
11014  027502                           MOVE    MEM,OUT1 <4>    ;GOOD DATA
11015           000061                  MICPC=MICPC+1
11016  027502   041224                  .WORD   .$.
11017  027504                           MOVE    BREG,OUT1 <5>   ;BAD DATA.
11018           000062                  MICPC=MICPC+1
11019  027504   061225                  .WORD   .$.
11020  027506                           MOVE    # 15,BREG       ;SET TYPE OF ERROR.
11021           000063                  MICPC=MICPC+1
11022  027506   000415                  .WORD   .$.
11023  027510                           MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
11024           000064                  MICPC=MICPC+1
11025  027510   061223                  .WORD   .$.
11026  027512                           MOVE    # 06,BREG       ;LOAD FUNCTION CODE...
11027           000065                  MICPC=MICPC+1
11028  027512   000406                  .WORD   .$.
11029  027514                           MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
11030           000066                  MICPC=MICPC+1
11031  027514   061227                  .WORD   .$.
11032  027516                           CALL    ERROR           ;ALU 2A W/C ERROR...
11033  027516                           MOVE    # <MICPC+3>,BREG
11034           000067                  MICPC=MICPC+1
11035  027516   000471                  .WORD   .$.
```

```
11036  027520                          SBR     ERDR
11037          000070                  MICPC=MICPC+1
11038  027520  104400                  .WORD   .$.
11039  027522                          MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
11040          000071                  MICPC=MICPC+1
11041  027522  070204                  .WORD   .$.
11042  027524                          $BR     2$              ;LOOP ON ERROR...
11043          000072                  MICPC=MICPC+1
11044  027524  100453                  .WORD   .$.
11045  027526                  3$:     CALL    SCP1
11046  027526                          MOVE    # <MICPC+3>,BREG
11047          000073                  MICPC=MICPC+1
11048  027526  000475                  .WORD   .$.
11049  027530                          $BR     SCP1
11050          000074                  MICPC=MICPC+1
11051  027530  104427                  .WORD   .$.
11052  027532                          MOVE    SPAD <4>,MLR    ;
11053          000075                  MICPC=MICPC+1
11054  027532  070204                  .WORD   .$.
11055  027534                          $BR     2$              ; SCOPE THE DATA....
11056          000076                  MICPC=MICPC+1
11057  027534  100453                  .WORD   .$.
11058  027536                  6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
11059          000077                  MICPC=MICPC+1
11060  027536  070204                  .WORD   .$.
11061  027540                          MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
11062          000100                  MICPC=MICPC+1
11063  027540  057220                  .WORD   .$.
11064  027542                          MOVE    # 377,BREG      ;
11065          000101                  MICPC=MICPC+1
11066  027542  000777                  .WORD   .$.
11067  027544                          $OOD    SPAD <2>,BREG   ;SET C BIT...
11068          000102                  MICPC=MICPC+1
11069  027544  060402                  .WORD   .$.!.DO
11070  027546                          $ROL    SPAD <0>,BR.SP,MARINC           ;SP 0 & BR = 2A W/C
11071          000103                  MICPC=MICPC+1
11072  027546  077540                  .WORD   .$.
11073  027550                          MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
11074          000104                  MICPC=MICPC+1
11075  027550  057223                  .WORD   .$.
11076  027552                          $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
11077
11078
11079  027552                          SUB2C   SPAD <0>,MEM,NOP
11080          000105                  MICPC=MICPC+1
11081  027552  040360                  .WORD   .$.
11082  027554                          BZ      9$
11083          000106                  MICPC=MICPC+1
11084  027554  101521                  .WORD   .$.
11085  027556                          MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
11086          000107                  MICPC=MICPC+1
11087  027556  041224                  .WORD   .$.
11088  027560                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
11089          000110                  MICPC=MICPC+1
11090  027560  061225                  .WORD   .$.
11091  027562                          MOVE    # 23,BREG               ;ERROR TYPE...
```

```
11092                   000111                  MICPC=MICPC+1
11093  027562           000423                  .WORD   .$.
11094  027564                                   MOVE    BREG,OUT1 <CSR3>        ;
11095                   000112                  MICPC=MICPC+1
11096  027564           061223                  .WORD   .$.
11097  027566                                   MOVE    # 06,BREG               ;ALU FUNCTION CODE.
11098                   000113                  MICPC=MICPC+1
11099  027566           000406                  .WORD   .$.
11100  027570                                   MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
11101                   000114                  MICPC=MICPC+1
11102  027570           061227                  .WORD   .$.
11103  027572                                   CALL    ERROR   ;REPORT ERROR...
11104  027572                                   MOVE    # <MICPC+3>,BREG
11105                   000115                  MICPC=MICPC+1
11106  027572           000517                  .WORD   .$.
11107  027574                                   SBR     ERROR
11108                   000116                  MICPC=MICPC+1
11109  027574           104400                  .WORD   .$.
11110  027576                                   MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
11111                   000117                  MICPC=MICPC+1
11112  027576           070204                  .WORD   .$.
11113  027600                                   SBR     6$                      ;LOOP ON ERROR...
11114                   000120                  MICPC=MICPC+1
11115  027600           100477                  .WORD   .$.
11116  027602                           9$:     CALL    SCP1            ;SCOPE THE ERROR...
11117  027602                                   MOVE    # <MICPC+3>,BREG
11118                   000121                  MICPC=MICPC+1
11119  027602           000523                  .WORD   .$.
11120  027604                                   SBR     SCP1
11121                   000122                  MICPC=MICPC+1
11122  027604           104427                  .WORD   .$.
11123  027606                                   MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
11124                   000123                  MICPC=MICPC+1
11125  027606           070204                  .WORD   .$.
11126  027610                                   SBR     6$                      ;SCOPE THE DATA...
11127                   000124                  MICPC=MICPC+1
11128  027610           100477                  .WORD   .$.
11129  027612                                   MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
11130                   000125                  MICPC=MICPC+1
11131  027612           000404                  .WORD   .$.
11132  027614                                   SADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
11133  027614           077004                  .WORD   .$.!MARINC!.DO
11134  027616                                   SDEC    SPAD <7>                ;IS IT DONE??
11135                   000126                  MICPC=MICPC+1
11136  027616           063167          .WORD   .$.!.DSP
11137  027620                                   BZ      4$                      ;YES, SCOPE THE TEST.
11138                   000127                  MICPC=MICPC+1
11139  027620           101532                  .WORD   .$.
11140  027622                                   SBR     2$                      ;DO, THE NEXT.
11141                   000130                  MICPC=MICPC+1
11142  027622           100453                  .WORD   .$.
11143  027624                           4$:     CALL    SCPE                    ;SCOPE THE TEST...
11144  027624                                   MOVE    # <MICPC+3>,BREG
11145                   000131                  MICPC=MICPC+1
11146  027624           000533                  .WORD   .$.
11147  027626                                   SBR     SCPE
```

```
11148            000132                          MICPC=MICPC+1
11149  027626    104454                          .WORD   .$.
11150  027630                                    $BR     1$              ;DO THE NEXT ITERATION...
11151            000133                          MICPC=MICPC+1
11152  027630    100400                          .WORD   .$.
11153  027632                            $ALUT1  0,<SUB>,SUB1C,0,0,-1,-1,1,1,0,0,0,0,125,125,253,253,0,0,<A-B>,1,16
11154  027632                            $XZ
11155
11156
11157                                             ;*************************** TEST 52 ****************************
11158                                             ;*ALU TEST
11159                                             ;*TEST OF ALU FUNCTION SUB WITH C BIT CLEARED.
11160                                             ;*TEST OF ALU FUNCTION SUB WITH C BIT SET.
11161                                             ;*ALU FUNCTION (A-B)
11162                                             ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
11163                                             ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
11164  027632                            $XZ
11165                                             ;:************************************************************************
11166
11167  027632                            $TSTN
11168                                             ;  TEST 52
11169                                             ;--------------
11170  027632  012737  000052  001202    TST52:  MOV     #52,$TSTNM              ; LOAD THE NO. OF THIS TEST
11171  027640  012737  030160  001442            MOV     #TST53,NEXT             ; POINT TO THE START OF NEXT TEST.
11172                                           ;R1 CONTAINS BASE KMC11 ADDRESS
11173  027646  004737  035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
11174  027652  027666                            MCT52
11175  027654  104022                            ERROR   22                      ;TIME OUT ERROR...
11176  027656  012706  001200                    MOV     #STACK,SP               ;RESET STACK...
11177  027662  000177  151554                    JMP     @NEXT                   ;GO TO NEXT TEST...
11178  027666                            MCT52:
11179  027666                            1$:     MOVE    # 0,MLR          ;SET MAR+LO.
11180            000000                          MICPC=MICPC+1
11181  027666    010000                          .WORD   .$.
11182  027670                                    MOVE    # 0,MPR          ;SET MAR+HI.
11183            000001                          MICPC=MICPC+1
11184  027670    004000                          .WORD   .$.
11185  027672                                    MOVE    # 0,BREG                 ;
11186            000002                          MICPC=MICPC+1
11187  027672    000400                          .WORD   .$.
11188  027674                                    MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
11189            000003                          MICPC=MICPC+1
11190  027674    063236                          .WORD   .$.
11191  027676                                    MOVE    BREG,SPAD <0>   ;
11192            000004                          MICPC=MICPC+1
11193  027676    063220                          .WORD   .$.
11194  027700                                    MOVE    BREG,SPAD <1>   ;
11195            000005                          MICPC=MICPC+1
11196  027700    063221                          .WORD   .$.
11197  027702                                    MOVE    BREG,SPAD <2>   ;
11198            000006                          MICPC=MICPC+1
11199  027702    063222                          .WORD   .$.
11200  027704                                    $DEC    $PAD <2>                ;
11201            000007                          MICPC=MICPC+1
11202  027704    063162                  .WORD   .$.!.DSP
11203  027706                                    MOVE    BREG,SPAD <4>           ;
```

```
11204                   000010          MICPC=MICPC+1
11205  027706  063224                   .WORD   .$.
11206  027710                           MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11207                   000011          MICPC=MICPC+1
11208  027710  016400                   .WORD   .$.
11209  027712                           MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
11210                   000012          MICPC=MICPC+1
11211  027712  016400                   .WORD   .$.
11212  027714                           MOVE    # 0,MEM MARINC
11213                   000013          MICPC=MICPC+1
11214  027714  016400                   .WORD   .$.
11215  027716                           MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11216                   000014          MICPC=MICPC+1
11217  027716  016400                   .WORD   .$.
11218  027720                           MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11219                   000015          MICPC=MICPC+1
11220  027720  016777                   .WORD   .$.
11221  027722                           MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11222                   000016          MICPC=MICPC+1
11223  027722  016400                   .WORD   .$.
11224  027724                           MOVE    # -1,MEM MARINC
11225                   000017          MICPC=MICPC+1
11226  027724  016777                   .WORD   .$.
11227  027726                           MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
11228                   000020          MICPC=MICPC+1
11229  027726  016777                   .WORD   .$.
11230  027730                           MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11231                   000021          MICPC=MICPC+1
11232  027730  016400                   .WORD   .$.
11233  027732                           MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11234                   000022          MICPC=MICPC+1
11235  027732  016777                   .WORD   .$.
11236  027734                           MOVE    # 1,MEM MARINC
11237                   000023          MICPC=MICPC+1
11238  027734  016401                   .WORD   .$.
11239  027736                           MOVE    # 1,MEM MARINC  ;RESULT WITH C BIT SET.
11240                   000024          MICPC=MICPC+1
11241  027736  016401                   .WORD   .$.
11242  027740                           MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
11243                   000025          MICPC=MICPC+1
11244  027740  016777                   .WORD   .$.
11245  027742                           MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11246                   000026          MICPC=MICPC+1
11247  027742  016777                   .WORD   .$.
11248  027744                           MOVE    # 0,MEM MARINC
11249                   000027          MICPC=MICPC+1
11250  027744  016400                   .WORD   .$.
11251  027746                           MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11252                   000030          MICPC=MICPC+1
11253  027746  016400                   .WORD   .$.
11254  027750                           MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11255                   000031          MICPC=MICPC+1
11256  027750  016525                   .WORD   .$.
11257  027752                           MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11258                   000032          MICPC=MICPC+1
11259  027752  016525                   .WORD   .$.
```

```
11260  027754
11261          000033
11262  027754  016400          MOVE    # 0,MEM MARINC
11263  027756                  MICPC=MICPC+1
11264          000034          .WORD   .S.
11265  027756  016400          MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11266  027760                  MICPC=MICPC+1
11267          000035          .WORD   .S.
11268  027760  016652          MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11269  027762                  MICPC=MICPC+1
11270          000036          .WORD   .S.
11271  027762  016525          MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11272  027764                  MICPC=MICPC+1
11273          000037          .WORD   .S.
11274  027764  016525          MOVE    # 125,MEM MARINC
11275  027766                  MICPC=MICPC+1
11276          000040          .WORD   .S.
11277  027766  016525          MOVE    # 125,MEM MARINC        ;RESULT WITH C BIT SET.
11278  027770                  MICPC=MICPC+1
11279          000041          .WORD   .S.
11280  027770  016525          MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11281  027772                  MICPC=MICPC+1
11282          000042          .WORD   .S.
11283  027772  016652          MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
11284  027774                  MICPC=MICPC+1
11285          000043          .WORD   .S.
11286  027774  016653          MOVE    # 253,MEM MARINC
11287  027776                  MICPC=MICPC+1
11288          000044          .WORD   .S.
11289  027776  016653          MOVE    # 253,MEM MARINC        ;RESULT WITH C BIT SET.
11290  030000                  MICPC=MICPC+1
11291          000045          .WORD   .S.
11292  030000  016652          MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11293  030002                  MICPC=MICPC+1
11294          000046          .WORD   .S.
11295  030002  016652          MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11296  030004                  MICPC=MICPC+1
11297          000047          .WORD   .S.
11298  030004  016400          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11299  030006                  MICPC=MICPC+1
11300          000050          .WORD   .S.
11301  030006  016400          MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11302  030010                  MICPC=MICPC+1
11303          000051          .WORD   .S.
11304  030010  003007          MOVE    # 7,SPAD <7>    ;SET THE COUNT.
11305                          MICPC=MICPC+1
11306  030012                  .WORD   .S.
11307          000052          MOVE    # 0,MLR         ;MAR←0.
11308  030012  010000          MICPC=MICPC+1
11309                          .WORD   .S.
11310  030014
11311          000053  2$:     MOVE    # 0,BREG                 ;
11312  030014  000400          MICPC=MICPC+1
11313  030016                  .WORD   .S.
11314          000054          $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
11315  030016  060401          MICPC=MICPC+1
                                .WORD   .S.!.DD
```

```
11316  030020                    MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
11317          000055            MICPC=MICPC+1
11318  030020  057220            .WORD   .$.
11319  030022                    $UBIC   MEM,SPAD <0>,BR.SP,MARINC       ;SPAD  <0>:=$FUNC.
11320          000056            MICPC=MICPC+1
11321  030022  057740            .WORD   .$.
11322  030024                    $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
11323
11324
11325  030024                    SUB2C   SPAD <0>,MEM,NOP
11326          000057            MICPC=MICPC+1
11327  030024  040360            .WORD   .$.
11328  030026                    BZ      3$
11329          000060            MICPC=MICPC+1
11330  030026  101473            .WORD   .$.
11331  030030                    MOVE    MEM,OUT1 <4>    ;GOOD DATA
11332          000061            MICPC=MICPC+1
11333  030030  041224            .WORD   .$.
11334  030032                    MOVE    BREG,OUT1 <5>   ;BAD DATA.
11335          000062            MICPC=MICPC+1
11336  030032  061225            .WORD   .$.
11337  030034                    MOVE    # 15,BREG       ;SET TYPE OF ERROR.
11338          000063            MICPC=MICPC+1
11339  030034  000415            .WORD   .$.
11340  030036                    MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
11341          000064            MICPC=MICPC+1
11342  030036  061223            .WORD   .$.
11343  030040                    MOVE    # 16,BREG       ;LOAD FUNCTION CODE...
11344          000065            MICPC=MICPC+1
11345  030040  000416            .WORD   .$.
11346  030042                    MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
11347          000066            MICPC=MICPC+1
11348  030042  061227            .WORD   .$.
11349  030044                    CALL    ERROR           ;ALU SUB ERROR...
11350  030044                    MOVE    # <MICPC+3>,BREG
11351          000067            MICPC=MICPC+1
11352  030044  000471            .WORD   .$.
11353  030046                    $BR     ERROR
11354          000070            MICPC=MICPC+1
11355  030046  104400            .WORD   .$.
11356  030050                    MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
11357          000071            MICPC=MICPC+1
11358  030050  070204            .WORD   .$.
11359  030052                    SBR     2$              ;LOOP ON ERROR...
11360          000072            MICPC=MICPC+1
11361  030052  100453            .WORD   .$.
11362  030054            3$:     CALL    SCP1
11363  030054                    MOVE    # <MICPC+3>,BREG
11364          000073            MICPC=MICPC+1
11365  030054  000475            .WORD   .$.
11366  030056                    $BR     SCP1
11367          000074            MICPC=MICPC+1
11368  030056  104427            .WORD   .$.
11369  030060                    MOVE    SPAD <4>,MLR    ;
11370          000075            MICPC=MICPC+1
11371  030060  070204            .WORD   .$.
```

```
11372  030062                          SBR     2S                      ; SCOPE THE DATA....
11373         000076                   MICPC=MICPC+1
11374  030062 100453                   .WORD   .S.
11375  030064                   6S:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
11376         000077                   MICPC=MICPC+1
11377  030064 070204                   .WORD   .S.
11378  030066                          MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
11379         000100                   MICPC=MICPC+1
11380  030066 057220                   .WORD   .S.
11381  030070                          MOVE    # 377,BREG      ;               .
11382         000101                   MICPC=MICPC+1
11383  030070 000777                   .WORD   .S.
11384  030072                          $COD    SPAD <2>,BREG   ;SET C BIT...
11385         000102                   MICPC=MICPC+1
11386  030072 060402                   .WORD   .S.!.DD
11387  030074                          SUB1C   MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = SUB
11388         000103                   MICPC=MICPC+1
11389  030074 057740                   .WORD   .S.
11390  030076                          MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
11391         000104                   MICPC=MICPC+1
11392  030076 057223                   .WORD   .S.
11393  030100                          $IFEQ   MEM,SPAD <0>    9S      ;BR IF GOOD...
11394
11395
11396  030100                          SUB2C   SPAD <0>,MEM,NOP
11397         000105                   MICPC=MICPC+1
11398  030100 040360                   .WORD   .S.
11399  030102                          BZ      9S
11400         000106                   MICPC=MICPC+1
11401  030102 101521                   .WORD   .S.
11402  030104                          MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
11403         000107                   MICPC=MICPC+1
11404  030104 041224                   .WORD   .S.
11405  030106                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
11406         000110                   MICPC=MICPC+1
11407  030106 061225                   .WORD   .S.
11408  030110                          MOVE    # 23,BREG               ;ERROR TYPE...
11409         000111                   MICPC=MICPC+1
11410  030110 000423                   .WORD   .S.
11411  030112                          MOVE    BREG,OUT1 <CSR3>        ;
11412         000112                   MICPC=MICPC+1
11413  030112 061223                   .WORD   .S.
11414  030114                          MOVE    # 16,BREG               ;ALU FUNCTION CODE.
11415         000113                   MICPC=MICPC+1
11416  030114 000416                   .WORD   .S.
11417  030116                          MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
11418         000114                   MICPC=MICPC+1
11419  030116 061227                   .WORD   .S.
11420  030120                          CALL    ERROR   ;REPORT ERROR...
11421  030120                          MOVE    # <MICPC+3>,BREG
11422         000115                   MICPC=MICPC+1
11423  030120 000517                   .WORD   .S.
11424  030122                          SBR     ERROR
11425         000116                   MICPC=MICPC+1
11426  030122 104400                   .WORD   .S.
11427  030124                          MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
```

```
11428  000117                    MICPC=MICPC+1
11429  030124 070204             .WORD   .$.
11430  030126                    $BR     6$                      ;LOOP ON ERROR...
11431  000120                    MICPC=MICPC+1
11432  030126 100477             .WORD   .$.
11433  030130                    CALL    SCP1                    ;SCOPE THE ERROR...
11434  030130                    MOVE    # <MICPC+3>,BREG
11435  000121                    MICPC=MICPC+1
11436  030130 000523             .WORD   .$.
11437  030132                    $BR     SCP1
11438  000122                    MICPC=MICPC+1
11439  030132 104427             .WORD   .$.
11440  030134                    MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER...
11441  000123                    MICPC=MICPC+1
11442  030134 070204             .WORD   .$.
11443  030136                    $BR     6$                      ;SCOPE THE DATA...
11444  000124                    MICPC=MICPC+1
11445  030136 100477             .WORD   .$.
11446  030140                    MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
11447  000125                    MICPC=MICPC+1
11448  030140 000404             .WORD   .$.
11449  030142                    $ADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
11450  030142 077004             .WORD   .$.!MARINC!.DO
11451  030144                    $DEC    SPAD <7>                ;IS IT DONE??
11452  000126                    MICPC=MICPC+1
11453  030144 063167      .WORD  .$.!.DSP
11454  030146                    BZ      4$                      ;YES, SCOPE THE TEST.
11455  000127                    MICPC=MICPC+1
11456  030146 101532             .WORD   .$.
11457  030150                    $BR     2$                      ;DO, THE NEXT.
11458  000130                    MICPC=MICPC+1
11459  030150 100453             .WORD   .$.
11460  030152            4$:     CALL    SCPE                    ;SCOPE THE TEST...
11461  030152                    MOVE    # <MICPC+3>,BREG
11462  000131                    MICPC=MICPC+1
11463  030152 000533             .WORD   .$.
11464  030154                    $BR     SCPE
11465  000132                    MICPC=MICPC+1
11466  030154 104454             .WORD   .$.
11467  030156                    $BR     1$                      ;DO THE NEXT ITERATION...
11468  000133                    MICPC=MICPC+1
11469  030156 100400             .WORD   .$.
11470  030160            $ALUT1  0,<ADD W/C>,ADDWC,0,1,-1,0,-1,0,376,-1,252,253,-1,0,-1,0,124,125,<A PLUS B PLUS
11471  030160            $XZ
11472
11473
11474
11475                             ;************************** TEST 53 **************************
11476                             ;*ALU TEST
11477                             ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT CLEARED.
11478                             ;*TEST OF ALU FUNCTION ADD W/C WITH C BIT SET.
11479                             ;*ALU FUNCTION (A PLUS B PLUS C)
11480                             ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
11481  030160            $XZ      ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
11482                             ;:**********************************************************************
11483
```

```
11484  030160                          $TSTN
11485                                          ;  TEST 53
11486                                          ;-------------
11487  030160  012737  000053  001202  TST53:  MOV     #53,$TSTNM          ; LOAD THE NO. OF THIS TEST
11488  030166  012737  030506  001442          MOV     #TST54,NEXT         ; POINT TO THE START OF NEXT TEST.
11489                                  ;R1 CONTAINS BASE KMC11 ADDRESS
11490  030174  004737  035536                  JSR     PC,LDVRWT           ;LOAD-VERIFY-WAIT.
11491  030200  030214                  MCT53
11492  030202  104022                  ERROR   22                          ;TIME OUT ERROR...
11493  030204  012706  001200          MOV     #STACK,SP                   ;RESET STACK...
11494  030210  000177  151226          JMP     @NEXT                       ;GO TO NEXT TEST...
11495  030214                  MCT53:
11496  030214                  1$:     MOVE    # 0,MLR            ;SET MAR<LO.
11497          000000                  MICPC=MICPC+1
11498  030214  010000                  .WORD   .$.
11499  030216                          MOVE    # 0,MPR            ;SET MAR<HI.
11500          000001                  MICPC=MICPC+1
11501  030216  004000                  .WORD   .$.
11502  030220                          MOVE    # 0,BREG                    ;
11503          000002                  MICPC=MICPC+1
11504  030220  000400                  .WORD   .$.
11505  030222                          MOVE    BREG,SPAD <16>             ;FOR RETURN ADDRESS...
11506          000003                  MICPC=MICPC+1
11507  030222  063236                  .WORD   .$.
11508  030224                          MOVE    BREG,SPAD <0>      ;
11509          000004                  MICPC=MICPC+1
11510  030224  063220                  .WORD   .$.
11511  030226                          MOVE    BREG,SPAD <1>      ;
11512          000005                  MICPC=MICPC+1
11513  030226  063221                  .WORD   .$.
11514  030230                          MOVE    BREG,SPAD <2>      ;
11515          000006                  MICPC=MICPC+1
11516  030230  063222                  .WORD   .$.
11517  030232                          SDEC    SPAD <2>                   ;
11518          000007                  MICPC=MICPC+1
11519  030232  063162          .WORD   .$.!.DSP
11520  030234                          MOVE    BREG,SPAD <4>              ;
11521          000010                  MICPC=MICPC+1
11522  030234  063224                  .WORD   .$.
11523  030236                          MOVE    # 0,MEM MARINC     ;LOAD THE DATA IN MEMORY.
11524          000011                  MICPC=MICPC+1
11525  030236  016400                  .WORD   .$.
11526  030240                          MOVE    # 0,MEM MARINC     ;LOAD THE DATA IN MEMORY
11527          000012                  MICPC=MICPC+1
11528  030240  016400                  .WORD   .$.
11529  030242                          MOVE    # 0,MEM MARINC
11530          000013                  MICPC=MICPC+1
11531  030242  016400                  .WORD   .$.
11532  030244                          MOVE    # 1,MEM MARINC     ;RESULT WITH C BIT SET.
11533          000014                  MICPC=MICPC+1
11534  030244  016401                  .WORD   .$.
11535  030246                          MOVE    # -1,MEM MARINC    ;LOAD THE DATA IN MEMORY.
11536          000015                  MICPC=MICPC+1
11537  030246  016777                  .WORD   .$.
11538  030250                          MOVE    # 0,MEM MARINC     ;LOAD THE DATA IN MEMORY.
11539          000016                  MICPC=MICPC+1
```

```
11540  030250  016400                .WORD   .S.
11541  030252                        MOVE    # -1,MEM MARINC
11542                  000017         MICPC=MICPC+1
11543  030252  016777                .WORD   .S.
11544  030254                        MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11545                  000020         MICPC=MICPC+1
11546  030254  016400                .WORD   .S.
11547  030256                        MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11548                  000021         MICPC=MICPC+1
11549  030256  016400                .WORD   .S.
11550  030260                        MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11551                  000022         MICPC=MICPC+1
11552  030260  016777                .WORD   .S.
11553  030262                        MOVE    # -1,MEM MARINC
11554                  000023         MICPC=MICPC+1
11555  030262  016777                .WORD   .S.
11556  030264                        MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11557                  000024         MICPC=MICPC+1
11558  030264  016400                .WORD   .S.
11559  030266                        MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
11560                  000025         MICPC=MICPC+1
11561  030266  016777                .WORD   .S.
11562  030270                        MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11563                  000026         MICPC=MICPC+1
11564  030270  016777                .WORD   .S.
11565  030272                        MOVE    # 376,MEM MARINC
11566                  000027         MICPC=MICPC+1
11567  030272  016776                .WORD   .S.
11568  030274                        MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
11569                  000030         MICPC=MICPC+1
11570  030274  016777                .WORD   .S.
11571  030276                        MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
11572                  000031         MICPC=MICPC+1
11573  030276  016525                .WORD   .S.
11574  030300                        MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
11575                  000032         MICPC=MICPC+1
11576  030300  016525                .WORD   .S.
11577  030302                        MOVE    # 252,MEM MARINC
11578                  000033         MICPC=MICPC+1
11579  030302  016652                .WORD   .S.
11580  030304                        MOVE    # 253,MEM MARINC         ;RESULT WITH C BIT SET.
11581                  000034         MICPC=MICPC+1
11582  030304  016653                .WORD   .S.
11583  030306                        MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
11584                  000035         MICPC=MICPC+1
11585  030306  016652                .WORD   .S.
11586  030310                        MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
11587                  000036         MICPC=MICPC+1
11588  030310  016525                .WORD   .S.
11589  030312                        MOVE    # -1,MEM MARINC
11590                  000037         MICPC=MICPC+1
11591  030312  016777                .WORD   .S.
11592  030314                        MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11593                  000040         MICPC=MICPC+1
11594  030314  016400                .WORD   .S.
11595  030316                        MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
```

```
11596                000041                  MICPC=MICPC+1
11597  030316        016525                  .WORD   .$.
11598  030320                                MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
11599                000042                  MICPC=MICPC+1
11600  030320        016652                  .WORD   .$.
11601  030322                                MOVE    # -1,MEM MARINC
11602                000043                  MICPC=MICPC+1
11603  030322        016777                  .WORD   .$.
11604  030324                                MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11605                000044                  MICPC=MICPC+1
11606  030324        016400                  .WORD   .$.
11607  030326                                MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11608                000045                  MICPC=MICPC+1
11609  030326        016652                  .WORD   .$.
11610  030330                                MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11611                000046                  MICPC=MICPC+1
11612  030330        016652                  .WORD   .$.
11613  030332                                MOVE    # 124,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11614                000047                  MICPC=MICPC+1
11615  030332        016524                  .WORD   .$.
11616  030334                                MOVE    # 125,MEM MARINC        ;RESULT WITH C BIT SET.
11617                000050                  MICPC=MICPC+1
11618  030334        016525                  .WORD   .$.
11619  030336                                MOVE    # 7,SPAD <7>    ;SET THE COUNT.
11620                000051                  MICPC=MICPC+1
11621  030336        003007                  .WORD   .$.
11622
11623  030340                                MOVE    # 0,MLR         ;MAR+0.
11624                000052                  MICPC=MICPC+1
11625  030340        010000                  .WORD   .$.
11626
11627  030342                        2$:     MOVE    # 0,BREG                 ;
11628                000053                  MICPC=MICPC+1
11629  030342        000400                  .WORD   .$.
11630  030344                                SADD    SPAD <1>,BREG   ;CLEAR C BIT.
11631                000054                  MICPC=MICPC+1
11632  030344        060401                  .WORD   .$.!.DD
11633  030346                                MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
11634                000055                  MICPC=MICPC+1
11635  030346        057220                  .WORD   .$.
11636  030350                                ADDWC   MEM,SPAD <0>,BR.SP,MARINC       ;SPAD  <0>:=$FUNC.
11637                000056                  MICPC=MICPC+1
11638  030350        057420                  .WORD   .$.
11639  030352                                SIFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
11640
11641
11642  030352                                SUB2C   SPAD <0>,MEM,NOP
11643                000057                  MICPC=MICPC+1
11644  030352        040360                  .WORD   .$.
11645  030354                                BZ      3$
11646                000060                  MICPC=MICPC+1
11647  030354        101473                  .WORD   .$.
11648  030356                                MOVE    MEM,OUT1 <4>    ;GOOD DATA
11649                000061                  MICPC=MICPC+1
11650  030356        041224                  .WORD   .$.
11651  030360                                MOVE    BREG,OUT1 <5>   ;BAD DATA.
```

```
11652                   000062          MICPC=MICPC+1
11653   030360  061225                  .WORD   .$.
11654   030362                          MOVE    # 15,BREG       ;SET TYPE OF ERROR.
11655                   000063          MICPC=MICPC+1
11656   030362  000415                  .WORD   .$.
11657   030364                          MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
11658                   000064          MICPC=MICPC+1
11659   030364  061223                  .WORD   .$.
11660   030366                          MOVE    # 01,BREG       ;LOAD FUNCTION CODE...
11661                   000065          MICPC=MICPC+1
11662   030366  000401                  .WORD   .$.
11663   030370                          MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
11664                   000066          MICPC=MICPC+1
11665   030370  061227                  .WORD   .$.
11666   030372                          CALL    ERROR           ;ALU ADD W/C ERROR...
11667   030372                          MOVE    # <MICPC+3>,BREG
11668                   000067          MICPC=MICPC+1
11669   030372  000471                  .WORD   .$.
11670   030374                          SBR     ERROR
11671                   000070          MICPC=MICPC+1
11672   030374  104400                  .WORD   .$.
11673   030376                          MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
11674                   000071          MICPC=MICPC+1
11675   030376  070204                  .WORD   .$.
11676   030400                          SBR     25              ;LOOP ON ERROR...
11677                   000072          MICPC=MICPC+1
11678   030400  100453                  .WORD   .$.
11679   030402                  35:     CALL    SCP1
11680   030402                          MOVE    # <MICPC+3>,BREG
11681                   000073          MICPC=MICPC+1
11682   030402  000475                  .WORD   .$.
11683   030404                          SBR     SCP1
11684                   000074          MICPC=MICPC+1
11685   030404  104427                  .WORD   .$.
11686   030406                          MOVE    SPAD <4>,MLR    ;
11687                   000075          MICPC=MICPC+1
11688   030406  070204                  .WORD   .$.
11689   030410                          SBR     25              ; SCOPE THE DATA....
11690                   000076          MICPC=MICPC+1
11691   030410  100453                  .WORD   .$.
11692   030412                  65:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
11693                   000077          MICPC=MICPC+1
11694   030412  070204                  .WORD   .$.
11695   030414                          MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
11696                   000100          MICPC=MICPC+1
11697   030414  057220                  .WORD   .$.
11698   030416                          MOVE    # 377,BREG      ;
11699                   000101          MICPC=MICPC+1
11700   030416  000777                  .WORD   .$.
11701   030420                          $ADD    SPAD <2>,BREG   ;SET C BIT...
11702                   000102          MICPC=MICPC+1
11703   030420  060402                  .WORD   .$.!.DD
11704   030422                          ADDWC   MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = ADD W/C
11705                   000103          MICPC=MICPC+1
11706   030422  057420                  .WORD   .$.
11707   030424                          MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
```

```
11708                 000104                  MICPC=MICPC+1
11709  030424  057223                         .WORD   .$.
11710  030426                                 $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
11711
11712
11713  030426                                 SUB2C   SPAD <0>,MEM,NOP
11714                 000105                  MICPC=MICPC+1
11715  030426  040360                         .WORD   .$.
11716  030430                                 BZ      9$
11717                 000106                  MICPC=MICPC+1
11718  030430  101521                         .WORD   .$.
11719  030432                                 MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
11720                 000107                  MICPC=MICPC+1
11721  030432  041224                         .WORD   .$.
11722  030434                                 MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
11723                 000110                  MICPC=MICPC+1
11724  030434  061225                         .WORD   .$.
11725  030436                                 MOVE    # 23,BREG               ;ERROR TYPE...
11726                 000111                  MICPC=MICPC+1
11727  030436  000423                         .WORD   .$.
11728  030440                                 MOVE    BREG,OUT1 <CSR3>        ;
11729                 000112                  MICPC=MICPC+1
11730  030440  061223                         .WORD   .$.
11731  030442                                 MOVE    # 01,BREG               ;ALU FUNCTION CODE.
11732                 000113                  MICPC=MICPC+1
11733  030442  000401                         .WORD   .$.
11734  030444                                 MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
11735                 000114                  MICPC=MICPC+1
11736  030444  061227                         .WORD   .$.
11737  030446                                 CALL    ERROR   ;REPORT ERROR...
11738  030446                                 MOVE    # <MICPC+3>,BREG
11739                 000115                  MICPC=MICPC+1
11740  030446  000517                         .WORD   .$.
11741  030450                                 SBR     ERROR
11742                 000116                  MICPC=MICPC+1
11743  030450  104400                         .WORD   .$.
11744  030452                                 MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
11745                 000117                  MICPC=MICPC+1
11746  030452  070204                         .WORD   .$.
11747  030454                                 SBR     6$                      ;LOOP ON ERROR...
11748                 000120                  MICPC=MIC*C+1
11749  030454  100477                         .WORD   .$.
11750  030456                         9$:     CALL    SCP1            ;SCOPE THE ERROR...
11751  030456                                 MOVE    # <MICPC+3>,BREG
11752                 000121                  MICPC=MICPC+1
11753  030456  000523                         .WORD   .$.
11754  030460                                 SBR     SCP1
11755                 000122                  MICPC=MICPC+1
11756  030460  104427                         .WORD   .$.
11757  030462                                 MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
11758                 000123                  MICPC=MICPC+1
11759  030462  070204                         .WORD   .$.
11760  030464                                 SBR     6$              ;SCOPE THE DATA...
11761                 000124                  MICPC=MICPC+1
11762  030464  100477                         .WORD   .$.
11763  030466                                 MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
```

```
11764           000125                          MICPC=MICPC+1
11765 030466    000404                          .WORD   .$.
11766 030470                                    $ADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
11767 030470    077004                          .WORD   .$.!MARINC!.DO
11768 030472                                    $DEC    SPAD <7>                ;IS IT DONE??
11769           000126                          MICPC=MICPC+1
11770 030472    063167                  .WORD   .$.!.DSP
11771 030474                                    BZ      4$                      ;YES, SCOPE THE TEST.
11772           000127                          MICPC=MICPC+1
11773 030474    101532                          .WORD   .$.
11774 030476                                    $BR     2$                      ;DO, THE NEXT.
11775           000130                          MICPC=MICPC+1
11776 030476    100453                          .WORD   .$.
11777 030500                            4$:     CALL    SCPE                    ;SCOPE THE TEST...
11778 030500                                    MOVE    @ <MICPC+3>,BREG
11779           000131                          MICPC=MICPC+1
11780 030500    000533                          .WORD   .$.
11781 030502                                    $BR     SCPE
11782           000132                          MICPC=MICPC+1
11783 030502    104454                          .WORD   .$.
11784 030504                                    $BR     1$                      ;DO THE NEXT ITERATION...
11785           000133                          MICPC=MICPC+1
11786 030504    100400                          .WORD   .$.
11787 030506                            $ALUT1  0,<SUB W/C>,SUBWC,-1,0,376,377,0,1,-1,0,-1,0,124,125,252,253,-1,0,<A-B-C>,1,02
11788 030506                            $XZ
11789
11790
11791                                           ;************************** TEST 54 ****************************
11792                                           ;*ALU TEST
11793                                           ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT CLEARED.
11794                                           ;*TEST OF ALU FUNCTION SUB W/C WITH C BIT SET.
11795                                           ;*ALU FUNCTION (A-B-C)
11796                                           ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
11797                                           ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
11798 030506                            $XZ     ;;**************************************************************
11799
11800
11801 030506                            $TSTN
11802
11803                                           ;  TEST 54
11804 030506 012737 000054 001202       TST54:  ;-------------
11805 030514 012737 031034 001442               MOV     #54,STSTNM              ; LOAD THE NO. OF THIS TEST
11806                                           MOV     #TST55,NEXT             ; POINT TO THE START OF NEXT TEST.
11807 030522 004737 035536                                              ;R1 CONTAINS BASE KMC11 ADDRESS
11808 030526 030542                             JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
11809 030530 104022                             MCT54
11810 030532 012706 001200                      ERROR   22                      ;TIME OUT ERROR...
11811 030536 000177 150700                      MOV     #STACK,SP               ;RESET STACK...
11812 030542                                    JMP     @NEXT                   ;GO TO NEXT TEST...
11813 030542                            MCT54:
11814           000000                  1$:     MOVE    # 0,MLR         ;SET MAR<LO.
11815 030542    010000                          MICPC=MICPC+1
11816 030544                                    .WORD   .$.
11817           000001                          MOVE    # 0,MPR         ;SET MAR<HI.
11818 030544    004000                          MICPC=MICPC+1
11819 030546                                    .WORD   .$.
                                                MOVE    # 0,BREG                 ;
```

```
11820                   000002                  MICPC=MICPC+1
11821  030546  000400                           .WORD   .$.
11822  030550                                   MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
11823                   000003                  MICPC=MICPC+1
11824  030550  063236                           .WORD   .$.
11825  030552                                   MOVE    BREG,SPAD <0>   ;
11826                   000004                  MICPC=MICPC+1
11827  030552  063220                           .WORD   .$.
11828  030554                                   MOVE    BREG,SPAD <1>   ;
11829                   000005                  MICPC=MICPC+1
11830  030554  063221                           .WORD   .$.
11831  030556                                   MOVE    BREG,SPAD <2>   ;
11832                   000006                  MICPC=MICPC+1
11833  030556  063222                           .WORD   .$.
11834  030560                                   SDEC    SPAD <2>                ;
11835                   000007                  MICPC=MICPC+1
11836  030560  063162           .WORD           .$.!.DSP
11837  030562                                   MOVE    BREG,SPAD <4>           ;
11838                   000010                  MICPC=MICPC+1
11839  030562  063224                           .WORD   .$.
11840  030564                                   MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11841                   000011                  MICPC=MICPC+1
11842  030564  016400                           .WORD   .$.
11843  030566                                   MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
11844                   000012                  MICPC=MICPC+1
11845  030566  016400                           .WORD   .$.
11846  030570                                   MOVE    # -1,MEM MARINC
11847                   000013                  MICPC=MICPC+1
11848  030570  016777                           .WORD   .$.
11849  030572                                   MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11850                   000014                  MICPC=MICPC+1
11851  030572  016400                           .WORD   .$.
11852  030574                                   MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11853                   000015                  MICPC=MICPC+1
11854  030574  016777                           .WORD   .$.
11855  030576                                   MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11856                   000016                  MICPC=MICPC+1
11857  030576  016400                           .WORD   .$.
11858  030600                                   MOVE    # 376,MEM MARINC
11859                   000017                  MICPC=MICPC+1
11860  030600  016776                           .WORD   .$.
11861  030602                                   MOVE    # 377,MEM MARINC        ;RESULT WITH C BIT SET.
11862                   000020                  MICPC=MICPC+1
11863  030602  016777                           .WORD   .$.
11864  030604                                   MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
11865                   000021                  MICPC=MICPC+1
11866  030604  016400                           .WORD   .$.
11867  030606                                   MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11868                   000022                  MICPC=MICPC+1
11869  030606  016777                           .WORD   .$.
11870  030610                                   MOVE    # 0,MEM MARINC
11871                   000023                  MICPC=MICPC+1
11872  030610  016400                           .WORD   .$.
11873  030612                                   MOVE    # 1,MEM MARINC  ;RESULT WITH C BIT SET.
11874                   000024                  MICPC=MICPC+1
11875  030612  016401                           .WORD   .$.
```

```
11876  030614                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
11877         000025           MICPC=MICPC+1
11878  030614 016777           .WORD   .$.
11879  030616                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11880         000026           MICPC=MICPC+1
11881  030616 016777           .WORD   .$.
11882  030620                  MOVE    # -1,MEM MARINC
11883         000027           MICPC=MICPC+1
11884  030620 016777           .WORD   .$.
11885  030622                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11886         000030           MICPC=MICPC+1
11887  030622 016400           .WORD   .$.
11888  030624                  MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11889         000031           MICPC=MICPC+1
11890  030624 016525           .WORD   .$.
11891  030626                  MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11892         000032           MICPC=MICPC+1
11893  030626 016525           .WORD   .$.
11894  030630                  MOVE    # -1,MEM MARINC
11895         000033           MICPC=MICPC+1
11896  030630 016777           .WORD   .$.
11897  030632                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11898         000034           MICPC=MICPC+1
11899  030632 016400           .WORD   .$.
11900  030634                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11901         000035           MICPC=MICPC+1
11902  030634 016652           .WORD   .$.
11903  030636                  MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11904         000036           MICPC=MICPC+1
11905  030636 016525           .WORD   .$.
11906  030640                  MOVE    # 124,MEM MARINC
11907         000037           MICPC=MICPC+1
11908  030640 016524           .WORD   .$.
11909  030642                  MOVE    # 125,MEM MARINC        ;RESULT WITH C BIT SET.
11910         000040           MICPC=MICPC+1
11911  030642 016525           .WORD   .$.
11912  030644                  MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11913         000041           MICPC=MICPC+1
11914  030644 016525           .WORD   .$.
11915  030646                  MOVE    # 252,MEM MARINC        ·LOAD THE DATA IN MEMORY
11916         000042           MICPC=MICPC+1
11917  030646 016652           .WORD   .$.
11918  030650                  MOVE    # 252,MEM MARINC
11919         000043           MICPC=MICPC+1
11920  030650 016652           .WORD   .$.
11921  030652                  MOVE    # 253,MEM MARINC        ;RESULT WITH C BIT SET.
11922         000044           MICPC=MICPC+1
11923  030652 016653           .WORD   .$.
11924  030654                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11925         000045           MICPC=MICPC+1
11926  030654 016652           .WORD   .$.
11927  030656                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
11928         000046           MICPC=MICPC+1
11929  030656 016652           .WORD   .$.
11930  030660                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
11931         000047           MICPC=MICPC+1
```

```
11932  030660  016777          .WORD   .S.
11933  030662                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
11934          000050          MICPC=MICPC+1
11935  030662  016400          .WORD   .S.
11936  030664                  MOVE    # 7,SPAD <7>    ;SET THE COUNT.
11937          000051          MICPC=MICPC+1
11938  030664  003007          .WORD   .S.
11939
11940  030666                  MOVE    # 0,MLR         ;MAR←0.
11941          000052          MICPC=MICPC+1
11942  030666  010000          .WORD   .S.
11943
11944  030670          2$:     MOVE    # 0,BREG                 ;
11945          000053          MICPC=MICPC+1
11946  030670  000400          .WORD   .S.
11947  030672                  $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
11948          000054          MICPC=MICPC+1
11949  030672  060401          .WORD   .S.!.DD
11950  030674                  MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
11951          000055          MICPC=MICPC+1
11952  030674  057220          .WORD   .S.
11953  030676                  SUBWC   MEM,SPAD <0>,BR.SP,MARINC       ;SPAD  <0>:=$FUNC.
11954          000056          MICPC=MICPC+1
11955  030676  057440          .WORD   .S.
11956  030700                  $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
11957
11958
11959  030700                  SUB2C   SPAD <0>,MEM,NOP
11960          000057          MICPC=MICPC+1
11961  030700  040360          .WORD   .S.
11962  030702                  BZ      3$
11963          000060          MICPC=MICPC+1
11964  030702  101473          .WORD   .S.
11965  030704                  MOVE    MEM,OUT1 <4>    ;GOOD DATA
11966          000061          MICPC=MICPC+1
11967  030704  041224          .WORD   .S.
11968  030706                  MOVE    BREG,OUT1 <5>   ;BAD DATA.
11969          000062          MICPC=MICPC+1
11970  030706  061225          .WORD   .S.
11971  030710                  MOVE    # 15,BREG       ;SET TYPE OF ERROR.
11972          000063          MICPC=MICPC+1
11973  030710  000415          .WORD   .S.
11974  030712                  MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
11975          000064          MICPC=MICPC+1
11976  030712  061223          .WORD   .S.
11977  030714                  MOVE    # 02,BREG       ;LOAD FUNCTION CODE...
11978          000065          MICPC=MICPC+1
11979  030714  000402          .WORD   .S.
11980  030716                  MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
11981          000066          MICPC=MICPC+1
11982  030716  061227          .WORD   .S.
11983  030720                  CALL    EROR                    ;ALU SUB W/C ERROR...
11984  030720                  MOVE    # <MICPC+3>,BREG
11985          000067          MICPC=MICPC+1
11986  030720  000471          .WORD   .S.
11987  030722                  SBR     EROR
```

```
11988                000070                  MICPC=MICPC+1
11989  030722        104400                  .WORD   .$.
11990  030724                                MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
11991                000071                  MICPC=MICPC+1
11992  030724        070204                  .WORD   .$.
11993  030726                                SBR     2$              ;LOOP ON ERROR...
11994                000072                  MICPC=MICPC+1
11995  030726        100453                  .WORD   .$.
11996  030730                        3$:     CALL    SCP1
11997  030730                                MOVE    # <MICPC+3>,BREG
11998                000073                  MICPC=MICPC+1
11999  030730        000475                  .WORD   .$.
12000  030732                                SBR     SCP1
12001                000074                  MICPC=MICPC+1
12002  030732        104427                  .WORD   .$.
12003  030734                                MOVE    SPAD <4>,MLR    ;
12004                000075                  MICPC=MICPC+1
12005  030734        070204                  .WORD   .$.
12006  030736                                SBR     2$              ; SCOPE THE DATA....
12007                000076                  MICPC=MICPC+1
12008  030736        100453                  .WORD   .$.
12009  030740                        6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
12010                000077                  MICPC=MICPC+1
12011  030740        070204                  .WORD   .$.
12012  030742                                MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
12013                000100                  MICPC=MICPC+1
12014  030742        057220                  .WORD   .$.
12015  030744                                MOVE    # 377,BREG      ;
12016                000101                  MICPC=MICPC+1
12017  030744        000777                  .WORD   .$.
12018  030746                                $ADD    SPAD <2>,BREG   ;SET C BIT...
12019                000102                  MICPC=MICPC+1
12020  030746        060402                  .WORD   .$.!.DD
12021  030750                                SUBWC   MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = SUB W/C
12022                000103                  MICPC=MICPC+1
12023  030750        057440                  .WORD   .$.
12024  030752                                MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
12025                000104                  MICPC=MICPC+1
12026  030752        057223                  .WORD   .$.
12027  030754                                $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
12028
12029
12030  030754                                SUB2C   SPAD <0>,MEM,NOP
12031                000105                  MICPC=MICPC+1
12032  030754        040360                  .WORD   .$.
12033  030756                                BZ      9$
12034                000106                  MICPC=MICPC+1
12035  030756        101521                  .WORD   .$.
12036  030760                                MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
12037                000107                  MICPC=MICPC+1
12038  030760        041224                  .WORD   .$.
12039  030762                                MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
12040                000110                  MICPC=MICPC+1
12041  030762        061225                  .WORD   .$.
12042  030764                                MOVE    # 23,BREG               ;ERROR TYPE...
12043                000111                  MICPC=MICPC+1
```

```
12044  030764  000423                    .WORD   .$.
12045  030766                            MOVE    BREG,OUT1 <CSR3>        ;
12046          000112                    MICPC=MICPC+1
12047  030766  061223                    .WORD   .$.
12048  030770                            MOVE    # D2,BREG               ;ALU FUNCTION CODE.
12049          000113                    MICPC=MICPC+1
12050  030770  000402                    .WORD   .$.
12051  030772                            MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
12052          000114                    MICPC=MICPC+1
12053  030772  061227                    .WORD   .$.
12054  030774                            CALL    ERROR   ;REPORT ERROR...
12055  030774                            MOVE    # <MICPC+3>,BREG
12056          000115                    MICPC=MICPC+1
12057  030774  000517                    .WORD   .$.
12058  030776                            SBR     ERROR
12059          000116                    MICPC=MICPC+1
12060  030776  104400                    .WORD   .$.
12061  031000                            MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
12062          000117                    MICPC=MICPC+1
12063  031000  070204                    .WORD   .$.
12064  031002                            SBR     6$                      ;LOOP ON ERROR...
12065          000120                    MICPC=MICPC+1
12066  031002  100477                    .WORD   .$.
12067  031004                    9$:     CALL    SCP1            ;SCOPE THE ERROR...
12068  031004                            MOVE    # <MICPC+3>,BREG
12069          000121                    MICPC=MICPC+1
12070  031004  000523                    .WORD   .$.
12071  031006                            SBR     SCP1
12072          000122                    MICPC=MICPC+1
12073  031006  104427                    .WORD   .$.
12074  031010                            MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
12075          000123                    MICPC=MICPC+1
12076  031010  070204                    .WORD   .$.
12077  031012                            SBR     6$              ;SCOPE THE DATA...
12078          000124                    MICPC=MICPC+1
12079  031012  100477                    .WORD   .$.
12080  031014                            MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
12081          000125                    MICPC=MICPC+1
12082  031014  000404                    .WORD   .$.
12083  031016                            SADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
12084  031016  077004                    .WORD   .$.!MARINC!.DD
12085  031020                            SDEC    SPAD <7>                ;IS IT DONE??
12086          000126                    MICPC=MICPC+1
12087  031020  063167            .WORD   .$.!.DSP
12088  031022                            BZ      4$              ;YES, SCOPE THE TEST.
12089          000127                    MICPC=MICPC+1
12090  031022  101532                    .WORD   .$.
12091  031024                            SBR     2$              ;DO, THE NEXT.
12092          000130                    MICPC=MICPC+1
12093  031024  100453                    .WORD   .$.
12094  031026                    4$:     CALL    SCPE            ;SCOPE THE TEST...
12095  031026                            MOVE    # <MICPC+3>,BREG
12096          000131                    MICPC=MICPC+1
12097  031026  000533                    .WORD   .$.
12098  031030                            SBR     SCPE
12099          000132                    MICPC=MICPC+1
```

```
12100 031030 104454                        .WORD   .$.
12101 031032                               SBR     1$              ;DO THE NEXT ITERATION...
12102        000133                        MICPC=MICPC+1
12103 031032 100400                        .WORD   $
12104 031034                       SALUT1  0,<INC A>,$INC,1,1,0,0,1,1,0,0,126,126,253,253,126,126,253,253,<A PLUS 1>,0,03
12105 031034                       SXZ
12106
12107
12108                                      ;************************** TEST 55 **************************
12109                                      ;*ALU TEST
12110                                      ;*TEST OF ALU FUNCTION INC A WITH C BIT CLEARED.
12111                                      ;*TEST OF ALU FUNCTION INC A WITH C BIT SET.
12112                                      ;*ALU FUNCTION (A PLUS 1)
12113                                      ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
12114                                      ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
12115 031034                       SXZ
12116                                      ;;**********************************************************************
12117
12118 031034                       STSTN
12119                                      ;  TEST 55
12120                                      ;-------------
12121 031034 012737 000055 001202  TST55:  MOV     #55,STSTNM              ; LOAD THE NO. OF THIS TEST
12122 031042 012737 031362 001442          MOV     #TST56,NEXT             ; POINT TO THE START OF NEXT TEST.
12123                                                              ;R1 CONTAINS BASE KMC11 ADDRESS
12124 031050 004737 035536                 JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
12125 031054 031070                        MCT55
12126 031056 104022                        ERROR   22                      ;TIME OUT ERROR...
12127 031060 012706 001200                 MOV     #STACK,SP               ;RESET STACK...
12128 031064 000177 150352                 JMP     @NEXT                   ;GO TO NEXT TEST...
12129 031070                       MCT55:
12130 031070                       1$:     MOVE    # 0,MLR          ;SET MAR+LO.
12131        000000                        MICPC=MICPC+1
12132 031070 010000                        .WORD   .$.
12133 031072                               MOVE    # 0,MPR          ;SET MAR+HI.
12134        000001                        MICPC=MICPC+1
12135 031072 004000                        .WORD   .$.
12136 031074                               MOVE    # 0,BREG                 ;
12137        000002                        MICPC=MICPC+1
12138 031074 000400                        .WORD   .$.
12139 031076                               MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
12140        000003                        MICPC=MICPC+1
12141 031076 063236                        .WORD   .$.
12142 031100                               MOVE    BREG,SPAD <0>   ;
12143        000004                        MICPC=MICPC+1
12144 031100 063220                        .WORD   .$.
12145 031102                               MOVE    BREG,SPAD <1>   ;
12146        000005                        MICPC=MICPC+1
12147 031102 063221                        .WORD   .$.
12148 031104                               MOVE    BREG,SPAD <2>   ;
12149        000006                        MICPC=MICPC+1
12150 031104 063222                        .WORD   .$.
12151 031106                               SDEC    SPAD <2>                ;
12152        000007                        MICPC=MICPC+1
12153 031106 063162                .WORD   .$.!.DSP
12154 031110                               MOVE    BREG,SPAD <4>           ;
12155        000010                        MICPC=MICPC+1
```

```
12156  031110  063224                  .WORD   .S.
12157  031112                          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
12158          000011                  MICPC=MICPC+1
12159  031112  016400                  .WORD   .S.
12160  031114                          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
12161          000012                  MICPC=MICPC+1
12162  031114  016400                  .WORD   .S.
12163  031116                          MOVE    # 1,MEM MARINC
12164          000013                  MICPC=MICPC+1
12165  031116  016401                  .WORD   .S.
12166  031120                          MOVE    # 1,MEM MARINC  ;RESULT WITH C BIT SET.
12167          000014                  MICPC=MICPC+1
12168  031120  016401                  .WORD   .S.
12169  031122                          MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
12170          000015                  MICPC=MICPC+1
12171  031122  016777                  .WORD   .S.
12172  031124                          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
12173          000016                  MICPC=MICPC+1
12174  031124  016400                  .WORD   .S.
12175  031126                          MOVE    # 0,MEM MARINC
12176          000017                  MICPC=MICPC+1
12177  031126  016400                  .WORD   .S.
12178  031130                          MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
12179          000020                  MICPC=MICPC+1
12180  031130  016400                  .WORD   .S.
12181  031132                          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
12182          000021                  MICPC=MICPC+1
12183  031132  016400                  .WORD   .S.
12184  031134                          MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
12185          000022                  MICPC=MICPC+1
12186  031134  016777                  .WORD   .S.
12187  031136                          MOVE    # 1,MEM MARINC
12188          000023                  MICPC=MICPC+1
12189  031136  016401                  .WORD   .S.
12190  031140                          MOVE    # 1,MEM MARINC  ;RESULT WITH C BIT SET.
12191          000024                  MICPC=MICPC+1
12192  031140  016401                  .WORD   .S.
12193  031142                          MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
12194          000025                  MICPC=MICPC+1
12195  031142  016777                  .WORD   .S.
12196  031144                          MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
12197          000026                  MICPC=MICPC+1
12198  031144  016777                  .WORD   .S.
12199  031146                          MOVE    # 0,MEM MARINC
12200          000027                  MICPC=MICPC+1
12201  031146  016400                  .WORD   .S.
12202  031150                          MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
12203          000030                  MICPC=MICPC+1
12204  031150  016400                  .WORD   .S.
12205  031152                          MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12206          000031                  MICPC=MICPC+1
12207  031152  016525                  .WORD   .S.
12208  031154                          MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12209          000032                  MICPC=MICPC+1
12210  031154  016525                  .WORD   .S.
12211  031156                          MOVE    # 126,MEM MARINC
```

```
12212                   000033                  MICPC=MICPC+1
12213  031156  016526                           .WORD   .$.
12214  031160                                   MOVE    # 126,MEM MARINC        ;RESULT WITH C BIT SET.
12215                   000034                  MICPC=MICPC+1
12216  031160  016526                           .WORD   .$.
12217  031162                                   MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12220                   000035                  MICPC=MICPC+1
12221  031162  016652                           .WORD   .$.
12222  031164                                   MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12223                   000036                  MICPC=MICPC+1
12224  031164  016525                           .WORD   .$.
12225  031166                                   MOVE    # 253,MEM MARINC
12226                   000037                  MICPC=MICPC+1
12227  031166  016653                           .WORD   .$.
12230  031170                                   MOVE    # 253,MEM MARINC        ;RESULT WITH C BIT SET.
12231                   000040                  MICPC=MICPC+1
12232  031170  016653                           .WORD   .$.
12233  031172                                   MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12234                   000041                  MICPC=MICPC+1
12235  031172  016525                           .WORD   .$.
12236  031174                                   MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
12237                   000042                  MICPC=MICPC+1
12240  031174  016652                           .WORD   .$.
12241  031176                                   MOVE    # 126,MEM MARINC
12242                   000043                  MICPC=MICPC+1
12243  031176  016526                           .WORD   .$.
12244  031200                                   MOVE    # 126,MEM MARINC        ;RESULT WITH C BIT SET.
12245                   000044                  MICPC=MICPC+1
12246  031200  016526                           .WORD   .$.
12247  031202                                   MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12250                   000045                  MICPC=MICPC+1
12251  031202  016652                           .WORD   .$.
12252  031204                                   MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12253                   000046                  MICPC=MICPC+1
12254  031204  016652                           .WORD   .$.
12255  031206                                   MOVE    # 253,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12256                   000047                  MICPC=MICPC+1
12257  031206  016653                           .WORD   .$.
12260  031210                                   MOVE    # 253,MEM MARINC        ;RESULT WITH C BIT SET.
12261                   000050                  MICPC=MICPC+1
12262  031210  016653                           .WORD   .$.
12263  031212                                   MOVE    # 7,SPAD <7>    ;SET THE COUNT.
12264                   000051                  MICPC=MICPC+1
12265  031212  003007                           .WORD   .$.
12266
12267  031214                                   MOVE    # 0,MLR          ;MAR=0.
12270                   000052                  MICPC=MICPC+1
12271  031214  010000                           .WORD   .$.
12272
12273  031216                           2$:     MOVE    # 0,BREG                 ;
12274                   000053                  MICPC=MICPC+1
12275  031216  000400                           .WORD   .$.
12276  031220                                   $ADD    SPAD <1>,BREG   ;CLEAR C BIT.
12277                   000054                  MICPC=MICPC+1
12300  031220  060401                           .WORD   .$.!.DD
12301  031222                                   MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
```

```
12268                   000055                  MICPC=MICPC+1
12269  031222  057220                           .WORD   .$.
12270  031224                                   $INC    SPAD <0>,BR.SP,MARINC   ;
12271                   000056                  MICPC=MICPC+1
12272  031224  077460                           .WORD   .$.
12273  031226                                   $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
12274
12275
12276  031226                                   SUB2C   SPAD <0>,MEM,NOP
12277                   000057                  MICPC=MICPC+1
12278  031226  040360                           .WORD   .$.
12279  031230                                   BZ      2$
12280                   000060                  MICPC=MICPC+1
12281  031230  101473                           .WORD   .$.
12282  031232                                   MOVE    MEM,OUT1 <4>    ;GOOD DATA
12283                   000061                  MICPC=MICPC+1
12284  031232  041224                           .WORD   .$.
12285  031234                                   MOVE    BREG,OUT1 <5>   ;BAD DATA.
12286                   000062                  MICPC=MICPC+1
12287  031234  061225                           .WORD   .$.
12288  031236                                   MOVE    # 15,BREG       ;SET TYPE OF ERROR.
12289                   000063                  MICPC=MICPC+1
12290  031236  000415                           .WORD   .$.
12291  031240                                   MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
12292                   000064                  MICPC=MICPC+1
12293  031240  061223                           .WORD   .$.
12294  031242                                   MOVE    # 03,BREG       ;LOAD FUNCTION CODE...
12295                   000065                  MICPC=MICPC+1
12296  031242  000403                           .WORD   .$.
12297  031244                                   MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
12298                   000066                  MICPC=MICPC+1
12299  031244  061227                           .WORD   .$.
12300  031246                                   CALL    ERROR           ;ALU INC A ERROR...
12301  031246                                   MOVE    # <MICPC+3>,BREG
12302                   000067                  MICPC=MICPC+1
12303  031246  000471                           .WORD   .$.
12304  031250                                   SBR     ERROR
12305                   000070                  MICPC=MICPC+1
12306  031250  104400                           .WORD   .$.
12307  031252                                   MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
12308                   000071                  MICPC=MICPC+1
12309  031252  070204                           .WORD   .$.
12310  031254                                   SBR     2$              ;LOOP ON ERROR...
12311                   000072                  MICPC=MICPC+1
12312  031254  100453                           .WORD   .$.
12313  031256                           3$:     CALL    SCP1
12314  031256                                   MOVE    # <MICPC+3>,BREG
12315                   000073                  MICPC=MICPC+1
12316  031256  000475                           .WORD   .$.
12317  031260                                   SBR     SCP1
12318                   000074                  MICPC=MICPC+1
12319  031260  104427                           .WORD   .$.
12320  031262                                   MOVE    SPAD <4>,MLR    ;
12321                   000075                  MICPC=MICPC+1
12322  031262  070204                           .WORD   .$.
12323  031264                                   SBR     2S              ; SCOPE THE DATA....
```

```
12324            000076                  MICPC=MICPC+1
12325  031264    100453                  .WORD   .$.
12326  031266                    6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
12327            000077                  MICPC=MICPC+1
12328  031266    070204                  .WORD   .$.
12329  031270                            MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
12330            000100                  MICPC=MICPC+1
12331  031270    057220                  .WORD   .$.
12332  031272                            MOVE    # 377,BREG      ;
12333            000101                  MICPC=MICPC+1
12334  031272    000777                  .WORD   .$.
12335  031274                            $ADD    SPAD <2>,BREG   ;SET C BIT...
12336            000102                  MICPC=MICPC+1
12337  031274    060402                  .WORD   .$.!.DD
12338  031276                            $INC    SPAD <0>,BR.SP,MARINC           ;SP 0 & BR = INC A
12339            000103                  MICPC=MICPC+1
12340  031276    077460                  .WORD   .$.
12341  031300                            MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
12342            000104                  MICPC=MICPC+1
12343  031300    057223                  .WORD   .$.
12344  031302                            $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
12345
12346
12347  031302                            SUB2C   SPAD <0>,MEM,NOP
12348            000105                  MICPC=MICPC+1
12349  031302    040360                  .WORD   .$.
12350  031304                            BZ      9$
12351            000106                  MICPC=MICPC+1
12352  031304    101521                  .WORD   .$.
12353  031306                            MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
12354            000107                  MICPC=MICPC+1
12355  031306    041224                  .WORD   .$.
12356  031310                            MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
12357            000110                  MICPC=MICPC+1
12358  031310    061225                  .WORD   .$.
12359  031312                            MOVE    # 23,BREG               ;ERROR TYPE...
12360            000111                  MICPC=MICPC+1
12361  031312    000423                  .WORD   .$.
12362  031314                            MOVE    BREG,OUT1 <CSR3>        ;
12363            000112                  MICPC=MICPC+1
12364  031314    061223                  .WORD   .$.
12365  031316                            MOVE    # 03,BREG               ;ALU FUNCTION CODE.
12366            000113                  MICPC=MICPC+1
12367  031316    000403                  .WORD   .$.
12368  031320                            MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
12369            000114                  MICPC=MICPC+1
12370  031320    061227                  .WORD   .$.
12371  031322                            CALL    ERROR   ;REPORT ERROR...
12372  031322                            MOVE    # <MICPC+3>,BREG
12373            000115                  MICPC=MICPC+1
12374  031322    000517                  .WORD   .$.
12375  031324                            SBR     ERROR
12376            000116                  MICPC=MICPC+1
12377  031324    104400                  .WORD   .$.
12378  031326                            MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
12379            000117                  MICPC=MICPC+1
```

```
12380  031326  070204                        .WORD   .$.
12381  031330                                $BR     6$                      ;LOOP ON ERROR...
12382          000120                        MICPC=MICPC+1
12383  031330  100477                        .WORD   .$.
12384  031332                        9$:     CALL    $CP1                    ;SCOPE THE ERROR...
12385  031332                                MOVE    # <MICPC+3>,BREG
12386          000121                        MICPC=MICPC+1
12387  031332  000523                        .WORD   .$.
12388  031334                                $BR     $CP1
12389          000122                        MICPC=MICPC+1
12390  031334  104427                        .WORD   .$.
12391  031336                                MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
12392          000123                        MICPC=MICPC+1
12393  031336  070204                        .WORD   .$.
12394  031340                                $BR     6$                      ;SCOPE THE DATA...
12395          000124                        MICPC=MICPC+1
12396  031340  100477                        .WORD   .$.
12397  031342                                MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
12398          000125                        MICPC=MICPC+1
12399  031342  000404                        .WORD   .$.
12400  031344                                $ADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
12401  031344  077004                        .WORD   .$.!MARINC!.DD
12402  031346                                $DEC    SPAD <7>                ;IS IT DONE??
12403          000126                        MICPC=MICPC+1
12404  031346  063167                .WORD   .$.!.DSP
12405  031350                                BZ      4$                      ;YES, SCOPE THE TEST.
12406          000127                        MICPC=MICPC+1
12407  031350  101532                        .WORD   .$.
12408  031352                                $BR     2$                      ;DO, THE NEXT.
12409          000130                        MICPC=MICPC+1
12410  031352  100453                        .WORD   .$.
12411  031354                        4$:     CALL    $CPE                    ;SCOPE THE TEST...
12412  031354                                MOVE    # <MICPC+3>,BREG
12413          000131                        MICPC=MICPC+1
12414  031354  000533                        .WORD   .$.
12415  031356                                $BR     $CPE
12416          000132                        MICPC=MICPC+1
12417  031356  104454                        .WORD   .$.
12418  031360                                $BR     1$                      ;DO THE NEXT ITERATION...
12419          000133                        MICPC=MICPC+1
12420  031360  100400                        .WORD   .$.
12421  031362                        $ALUT1  0,<2A>,$ASL,0,0,376,376,0,0,376,376,252,252,124,124,252,252,124,124,<A PLUS A>,0
12422  031362                        $XZ
12423
12424
12425                                 ;************************** TEST 56 **************************
12426                                 ;*ALU TEST
12427                                 ;*TEST OF ALU FUNCTION 2A WITH C BIT CLEARED.
12428                                 ;*TEST OF ALU FUNCTION 2A WITH C BIT SET.
12429                                 ;*ALU FUNCTION (A PLUS A)
12430                                 ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
12431                                 ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
12432  031362                        $XZ
12433                                 ;;*************************************************************
12434
12435  031362                        $TSTN
```

```
12436                                                   ;  TEST 56
12437                                                   ;--------------
12438  031362  012737  000056  001202  TST56:  MOV     #56,$TSTNM              ; LOAD THE NO. OF THIS TEST
12439  031370  012737  031710  001442          MOV     #TST57,NEXT             ; POINT TO THE START OF NEXT TEST.
12440                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
12441  031376  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
12442  031402  031416                          MCT56
12443  031404  104022                          ERROR   22                      ;TIME OUT ERROR...
12444  031406  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
12445  031412  000177  150024                  JMP     @NEXT                   ;GO TO NEXT TEST...
12446  031416                          MCT56:
12447  031416                          1$:     MOVE    # 0,MLR          ;SET MAR<-LO.
12448          000000                          MICPC=MICPC+1
12449  031416  010000                          .WORD   .$.
12450  031420                          MOVE    # 0,MPR          ;SET MAR<-HI.
12451          000001                          MICPC=MICPC+1
12452  031420  004000                          .WORD   .$.
12453  031422                          MOVE    # 0,BREG                 ;
12454          000002                          MICPC=MICPC+1
12455  031422  000400                          .WORD   .$.
12456  031424                          MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS...
12457          000003                          MICPC=MICPC+1
12458  031424  063236                          .WORD   .$.
12459  031426                          MOVE    BREG,SPAD (0)   ;
12460          000004                          MICPC=MICPC+1
12461  031426  063220                          .WORD   .$.
12462  031430                          MOVE    BREG,SPAD (1)   ;
12463          000005                          MICPC=MICPC+1
12464  031430  063221                          .WORD   .$.
12465  031432                          MOVE    BREG,SPAD (2)   ;
12466          000006                          MICPC=MICPC+1
12467  031432  063222                          .WORD   .$.
12468  031434                          SDEC    SPAD (2)                ;
12469          000007                          MICPC=MICPC+1
12470  031434  063162          .WORD   .$.!.DSP
12471  031436                          MOVE    BREG,SPAD (4)           ;
12472          000010                          MICPC=MICPC+1
12473  031436  063224                          .WORD   .$.
12474  031440                          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
12475          000011                          MICPC=MICPC+1
12476  031440  016400                          .WORD   .$.
12477  031442                          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
12478          000012                          MICPC=MICPC+1
12479  031442  016400                          .WORD   .$.
12480  031444                          MOVE    # 0,MEM MARINC
12481          000013                          MICPC=MICPC+1
12482  031444  016400                          .WORD   .$.
12483  031446                          MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
12484          000014                          MICPC=MICPC+1
12485  031446  016400                          .WORD   .$.
12486  031450                          MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
12487          000015                          MICPC=MICPC+1
12488  031450  016777                          .WORD   .$.
12489  031452                          MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
12490          000016                          MICPC=MICPC+1
12491  031452  016400                          .WORD   .$.
```

```
12492 031454
12493         000017
12494 031454  016776
12495 031456
12496         000020
12497 031456  016776
12498 031460
12499         000021
12500 031460  016400
12501 031462
12502         000022
12503 031462  016777
12504 031464
12505         000023
12506 031464  016400
12507 031466
12508         000024
12509 031466  016400
12510 031470
12511         000025
12512 031470  016777
12513 031472
12514         000026
12515 031472  016777
12516 031474
12517         000027
12518 031474  016776
12519 031476
12520         000030
12521 031476  016776
12522 031500
12523         000031
12524 031500  016525
12525 031502
12526         000032
12527 031502  016525
12528 031504
12529         000033
12530 031504  016652
12531 031506
12532         000034
12533 031506  016652
12534 031510
12535         000035
12536 031510  016652
12537 031512
12538         000036
12539 031512  016525
12540 031514
12541         000037
12542 031514  016524
12543 031516
12544         000040
12545 031516  016524
12546 031520
12547         000041
```

```
MOVE    # 376,MEM MARINC
MICPC=MICPC+1
.WORD   .S.
MOVE    # 376,MEM MARINC         ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 0,MEM MARINC   ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 0,MEM MARINC
MICPC=MICPC+1
.WORD   .S.
MOVE    # 0,MEM MARINC   ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .S.
MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
MICPC=MICPC+1
.WORD   .S.
MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 376,MEM MARINC
MICPC=MICPC+1
.WORD   .S.
MOVE    # 376,MEM MARINC         ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 252,MEM MARINC
MICPC=MICPC+1
.WORD   .S.
MOVE    # 252,MEM MARINC         ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 124,MEM MARINC
MICPC=MICPC+1
.WORD   .S.
MOVE    # 124,MEM MARINC         ;RESULT WITH C BIT SET.
MICPC=MICPC+1
.WORD   .S.
MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
MICPC=MICPC+1
```

```
12548  031520  016525                .WORD   .$.
12549  031522                        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
12550          000042                 MICPC=MICPC+1
12551  031522  016652                .WORD   .$.
12552  031524                        MOVE    # 252,MEM MARINC
12553          000043                 MICPC=MICPC+1
12554  031524  016652                .WORD   .$.
12555  031526                        MOVE    # 252,MEM MARINC        ;RESULT WITH C BIT SET.
12556          000044                 MICPC=MICPC+1
12557  031526  016652                .WORD   .$.
12558  031530                        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12559          000045                 MICPC=MICPC+1
12560  031530  016652                .WORD   .$.
12561  031532                        MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12562          000046                 MICPC=MICPC+1
12563  031532  016652                .WORD   .$.
12564  031534                        MOVE    # 124,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12565          000047                 MICPC=MICPC+1
12566  031534  016524                .WORD   .$.
12567  031536                        MOVE    # 124,MEM MARINC        ;RESULT WITH C BIT SET.
12568          000050                 MICPC=MICPC+1
12569  031536  016524                .WORD   .$.
12570  031540                        MOVE    # 7,SPAD <7>     ;SET THE COUNT.
12571          000051                 MICPC=MICPC+1
12572  031540  003007                .WORD   .$.
12573
12574  031542                        MOVE    # 0,MLR          ;MAR+0.
12575          000052                 MICPC=MICPC+1
12576  031542  010000                .WORD   .$.
12577
12578  031544                2$:     MOVE    # 0,BREG                  ;
12579          000053                 MICPC=MICPC+1
12580  031544  000400                .WORD   .$.
12581  031546                        SADD    SPAD <1>,BREG   ;CLEAR C BIT.
12582          000054                 MICPC=MICPC+1
12583  031546  060401                .WORD   .$.!.DD
12584  031550                        MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
12585          000055                 MICPC=MICPC+1
12586  031550  057220                .WORD   .$.
12587  031552                        SASL    SPAD <0>,BR.SP,MARINC   ;
12588          000056                 MICPC=MICPC+1
12589  031552  077520                .WORD   .$.
12590  031554                        SIFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
12591
12592
12593  031554                        SUB2C   SPAD <0>,MEM,NOP
12594          000057                 MICPC=MICPC+1
12595  031554  040360                .WORD   .$.
12596  031556                        BZ      3$
12597          000060                 MICPC=MICPC+1
12598  031556  101473                .WORD   .$.
12599  031560                        MOVE    MEM,OUT1 <4>    ;GOOD DATA
12600          000061                 MICPC=MICPC+1
12601  031560  041224                .WORD   .$.
12602  031562                        MOVE    BREG,OUT1 <5>   ;BAD DATA.
12603          000062                 MICPC=MICPC+1
```

```
12604  031562  061225                    .WORD   .$.
12605  031564                            MOVE    # 15,BREG       ;SET TYPE OF ERROR.
12606          000063                    MICPC=MICPC+1
12607  031564  000415                    .WORD   .$.
12608  031566                            MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
12609          000064                    MICPC=MICPC+1
12610  031566  061223                    .WORD   .$.
12611  031570                            MOVE    # 05,BREG       ;LOAD FUNCTION CODE...
12612          000065                    MICPC=MICPC+1
12613  031570  000405                    .WORD   .$.
12614  031572                            MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
12615          000066                    MICPC=MICPC+1
12616  031572  061227                    .WORD   .$.
12617  031574                            CALL    ERROR           ;ALU 2A ERROR...
12618  031574                            MOVE    # <MICPC+3>,BREG
12619          000067                    MICPC=MICPC+1
12620  031574  000471                    .WORD   .$.
12621  031576                            SBR     ERROR
12622          000070                    MICPC=MICPC+1
12623  031576  104400                    .WORD   .$.
12624  031600                            MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
12625          000071                    MICPC=MICPC+1
12626  031600  070204                    .WORD   .$.
12627  031602                            SBR     2S              ;LOOP ON ERROR...
12628          000072                    MICPC=MICPC+1
12629  031602  100453                    .WORD   .$.
12630  031604                    3S:     CALL    SCP1
12631  031604                            MOVE    # <MICPC+3>,BREG
12632          000073                    MICPC=MICPC+1
12633  031604  000475                    .WORD   .$.
12634  031606                            SBR     SCP1
12635          000074                    MICPC=MICPC+1
12636  031606  104427                    .WORD   .$.
12637  031610                            MOVE    SPAD <4>,MLR    ;
12638          000075                    MICPC=MICPC+1
12639  031610  070204                    .WORD   .$.
12640  031612                            SBR     2S              ; SCOPE THE DATA....
12641          000076                    MICPC=MICPC+1
12642  031612  100453                    .WORD   .$.
12643  031614                    6S:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
12644          000077                    MICPC=MICPC+1
12645  031614  070204                    .WORD   .$.
12646  031616                            MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
12647          000100                    MICPC=MICPC+1
12648  031616  057220                    .WORD   .$.
12649  031620                            MOVE    # 377,BREG      ;
12650          000101                    MICPC=MICPC+1
12651  031620  000777                    .WORD   .$.
12652  031622                            SADD    SPAD <2>,BREG   ;SET C BIT...
12653          000102                    MICPC=MICPC+1
12654  031622  060402                    .WORD   .$.!.D(
12655  031624                            SASL    SPAD <0>,BR.SP,MARINC           ;SP 0 & BR = 2A
12656          000103                    MICPC=MICPC+1
12657  031624  077520                    .WORD   .$.
12658  031626                            MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
12659          000104                    MICPC=MICPC+1
```

```
12660  031626  057223                  .WORD   .$.
12661  031630                          $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
12662
12663
12664  031630                          SUB2C   SPAD <0>,MEM,NOP
12665          000105                  MICPC=MICPC+1
12666  031630  040360                  .WORD   .$.
12667  031632                          BZ      9$
12670          000106                  MICPC=MICPC+1
12669  031632  101521                  .WORD   .$.
12670  031634                          MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
12671          000107                  MICPC=MICPC+1
12672  031634  041224                  .WORD   .$.
12673  031636                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
12674          000110                  MICPC=MICPC+1
12675  031636  061225                  .WORD   .$.
12676  031640                          MOVE    # 23,BREG               ;ERROR TYPE...
12677          000111                  MICPC=MICPC+1
12678  031640  000423                  .WORD   .$.
12679  031642                          MOVE    BREG,OUT1 <CSR3>        ;
12680          000112                  MICPC=MICPC+1
12681  031642  061223                  .WORD   .$.
12682  031644                          MOVE    # 05,BREG               ;ALU FUNCTION CODE.
12683          000113                  MICPC=MICPC+1
12684  031644  000405                  .WORD   .$.
12685  031646                          MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
12686          000114                  MICPC=MICPC+1
12687  031646  061227                  .WORD   .$.
12688  031650                          CALL    ERROR   ;REPORT ERROR...
12689  031650                          MOVE    # <MICPC+3>,BREG
12690          000115                  MICPC=MICPC+1
12691  031650  000517                  .WORD   .$.
12692  031652                          SBR     ERROR
12693          000116                  MICPC=MICPC+1
12694  031652  104400                  .WORD   .$.
12695  031654                          MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
12696          000117                  MICPC=MICPC+1
12697  031654  070204                  .WORD   .$.
12698  031656                          SBR     6$                      ;LOOP ON ERROR...
12699          000120                  MICPC=MICPC+1
12700  031656  100477                  .WORD   .$.
12701  031660                  9$:     CALL    SCP1            ;SCOPE THE ERROR...
12702  031660                          MOVE    # <MICPC+3>,BREG
12703          000121                  MICPC=MICPC+1
12704  031660  000523                  .WORD   .$.
12705  031662                          SBR     SCP1
12706          000122                  MICPC=MICPC+1
12707  031662  104427                  .WORD   .$.
12708  031664                          MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
12709          000123                  MICPC=MICPC+1
12710  031664  070204                  .WORD   .$.
12711  031666                          SBR     6$              ;SCOPE THE DATA...
12712          000124                  MICPC=MICPC+1
12713  031666  100477                  .WORD   .$.
12714  031670                          MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
12715          000125                  MICPC=MICPC+1
```

```
12716  031670  000404                        .WORD   .$.
12717  031672                                $ADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
12718  031672  077004                        .WORD   .$.!MARINC!.DO
12719  031674                                $DEC    SPAD <7>                ;IS IT DONE??
12720          000126                        MICPC=MICPC+1
12721  031674  063167              .WORD     .$.!.DSP
12722  031676                                BZ      4$                      ;YES, SCOPE THE TEST.
12723          000127                        MICPC=MICPC+1
12724  031676  101532                        .WORD   .$.
12725  031700                                $BR     2$                      ;DO, THE NEXT.
12726          000130                        MICPC=MICPC+1
12727  031700  100453                        .WORD   .$.
12728  031702                        4$:     CALL    SCPE                    ;SCOPE THE TEST...
12729  031702                                MOVE    # <MICPC+3>,BREG
12730          000131                        MICPC=MICPC+1
12731  031702  000533                        .WORD   .$.
12732  031704                                $BR     SCPE
12733          000132                        MICPC=MICPC+1
12734  031704  104454                        .WORD   .$.
12735  031706                                $BR     1$                      ;DO THE NEXT ITERATION...
12736          000133                        MICPC=MICPC+1
12737  031706  100400                        .WORD   .$.
12738  031710                        SALUT1  0,<A PLUS C>,$ADC,0,1,-1,0,0,1,-1,0,125,126,252,253,125,126,252,253,<A PLUS C>,0
12739  031710                        $XZ
12740
12741
12742                                        ;************************** TEST 57 **************************
12743                                        ;*ALU TEST
12744                                        ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT CLEARED.
12745                                        ;*TEST OF ALU FUNCTION A PLUS C WITH C BIT SET.
12746                                        ;*ALU FUNCTION (A PLUS C)
12747                                        ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
12748                                        ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
12749  031710                        $XZ
12750                                        ;:************************************************************************
12751
12752  031710                        $TSTN
12753                                        ;  TEST 57
12754                                        ;--------------
12755  031710  012737  000057  001202  TST57: MOV     #57,$TSTNM              ; LOAD THE NO. OF THIS TEST
12756  031716  012737  032236  001442        MOV     #TST60,NEXT             ; POINT TO THE START OF NEXT TEST.
12757                                                                ;R1 CONTAINS BASE KMC11 ADDRESS
12758  031724  004737  035536                JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
12759  031730  031744                        MCT57
12760  031732  104022                        ERROR   22                      ;TIME OUT ERROR...
12761  031734  012706  001200                MOV     #STACK,SP               ;RESET STACK...
12762  031740  000177  147476                JMP     @NEXT                   ;GO TO NEXT TEST...
12763  031744                        MCT57:
12764  031744                        1$:     MOVE    # 0,MLR          ;SET MAR<LO.
12765          000000                        MICPC=MICPC+1
12766  031744  010000                        .WORD   .$.
12767  031746                                MOVE    # 0,MPR          ;SET MAR<HI.
12768          000001                        MICPC=MICPC+1
12769  031746  004000                        .WORD   .$.
12770  031750                                MOVE    # 0,BREG                 ;
12771          000002                        MICPC=MICPC+1
```

```
12772 031750 000400                    .WORD   .$.
12773 031752                           MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
12774        000003                    MICPC=MICPC+1
12775 031752 063236                    .WORD   .$.
12776 031754                           MOVE    BREG,SPAD <0>   ;
12777        000004                    MICPC=MICPC+1
12778 031754 063220                    .WORD   .$.
12779 031756                           MOVE    BREG,SPAD <1>   ;
12780        000005                    MICPC=MICPC+1
12781 031756 063221                    .WORD   .$.
12782 031760                           MOVE    BREG,SPAD <2>   ;
12783        000006                    MICPC=MICPC+1
12784 031760 063222                    .WORD   .$.
12785 031762                           SDEC    SPAD <2>                ;
12786        000007                    MICPC=MICPC+1
12787 031762 063162            .WORD   .$.!.DSP
12788 031764                           MOVE    BREG,SPAD <4>           ;
12789        000010                    MICPC=MICPC+1
12790 031764 063224                    .WORD   .$.
12791 031766                           MOVE    #0,MEM MARINC   ;LOAD THE DATA IN MEMORY.
12792        000011                    MICPC=MICPC+1
12793 031766 016400                    .WORD   .$.
12794 031770                           MOVE    #0,MEM MARINC   ;LOAD THE DATA IN MEMORY
12795        000012                    MICPC=MICPC+1
12796 031770 016400                    .WORD   .$.
12797 031772                           MOVE    #0,MEM MARINC
12798        000013                    MICPC=MICPC+1
12799 031772 016400                    .WORD   .$.
12800 031774                           MOVE    #1,MEM MARINC   ;RESULT WITH C BIT SET.
12801        000014                    MICPC=MICPC+1
12802 031774 016401                    .WORD   .$.
12803 031776                           MOVE    #-1,MEM MARINC  ;LOAD THE DATA IN MEMORY.
12804        000015                    MICPC=MICPC+1
12805 031776 016777                    .WORD   .$.
12806 032000                           MOVE    #0,MEM MARINC   ;LOAD THE DATA IN MEMORY.
12807        000016                    MICPC=MICPC+1
12808 032000 016400                    .WORD   .$.
12809 032002                           MOVE    #-1,MEM MARINC
12810        000017                    MICPC=MICPC+1
12811 032002 016777                    .WORD   .$.
12812 032004                           MOVE    #0,MEM MARINC   ;RESULT WITH C BIT SET.
12813        000020                    MICPC=MICPC+1
12814 032004 016400                    .WORD   .$.
12815 032006                           MOVE    #0,MEM MARINC   ;LOAD THE DATA IN MEMORY.
12816        000021                    MICPC=MICPC+1
12817 032006 016400                    .WORD   .$.
12818 032010                           MOVE    #-1,MEM MARINC  ;LOAD THE DATA IN MEMORY.
12819        000022                    MICPC=MICPC+1
12820 032010 016777                    .WORD   .$.
12821 032012                           MOVE    #0,MEM MARINC
12822        000023                    MICPC=MICPC+1
12823 032012 016400                    .WORD   .$.
12824 032014                           MOVE    #1,MEM MARINC   ;RESULT WITH C BIT SET.
12825        000024                    MICPC=MICPC+1
12826 032014 016401                    .WORD   .$.
12827 032016                           MOVE    #-1,MEM MARINC  ;LOAD THE DATA IN MEMORY
```

```
12828  000025
12829  032016  016777          MICPC=MICPC+1
12830  032020                  .WORD   .S.
                               MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
12831  000026                  MICPC=MICPC+1
12832  032020  016777          .WORD   .S.
12833  032022                  MOVE    # -1,MEM MARINC
12834  000027                  MICPC=MICPC+1
12835  032022  016777          .WORD   .S.
12836  032024                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
12837  000030                  MICPC=MICPC+1
12838  032024  016400          .WORD   .S.
12839  032026                  MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12840  000031                  MICPC=MICPC+1
12841  032026  016525          .WORD   .S.
12842  032030                  MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12843  000032                  MICPC=MICPC+1
12844  032030  016525          .WORD   .S.
12845  032032                  MOVE    # 125,MEM MARINC
12846  000033                  MICPC=MICPC+1
12847  032032  016525          .WORD   .S.
12848  032034                  MOVE    # 126,MEM MARINC        ;RESULT WITH C BIT SET.
12849  000034                  MICPC=MICPC+1
12850  032034  016526          .WORD   .S.
12851  032036                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12852  000035                  MICPC=MICPC+1
12853  032036  016652          .WORD   .S.
12854  032040                  MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12855  000036                  MICPC=MICPC+1
12856  032040  016525          .WORD   .S.
12857  032042                  MOVE    # 252,MEM MARINC
12858  000037                  MICPC=MICPC+1
12859  032042  016652          .WORD   .S.
12860  032044                  MOVE    # 253,MEM MARINC        ;RESULT WITH C BIT SET.
12861  000040                  MICPC=MICPC+1
12862  032044  016653          .WORD   .S.
12863  032046                  MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12864  000041                  MICPC=MICPC+1
12865  032046  016525          .WORD   .S.
12866  032050                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY
12867  000042                  MICPC=MICPC+1
12868  032050  016652          .WORD   .S.
12869  032052                  MOVE    # 125,MEM MARINC
12870  000043                  MICPC=MICPC+1
12871  032052  016525          .WORD   .S.
12872  032054                  MOVE    # 126,MEM MARINC        ;RESULT WITH C BIT SET.
12873  000044                  MICPC=MICPC+1
12874  032054  016526          .WORD   .S.
12875  032056                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12876  000045                  MICPC=MICPC+1
12877  032056  016652          .WORD   .S.
12878  032060                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12879  000046                  MICPC=MICPC+1
12880  032060  016652          .WORD   .S.
12881  032062                  MOVE    # 252,MEM MARINC        ;LOAD THE DATA IN MEMORY.
12882  000047                  MICPC=MICPC+1
12883  032062  016652          .WORD   .S.
```

```
12884  032064                          MOVE    # 253,MEM MARINC         ;RESULT WITH C BIT SET.
12885          000050                  MICPC=MICPC+1
12886  032064  016653                  .WORD   .$.
12887  032066                          MOVE    # 7,SPAD <7>     ;SET THE COUNT.
12888          000051                  MICPC=MICPC+1
12889  032066  003007                  .WORD   .$.
12890
12891  032070                          MOVE    # 0,MLR          ;MAR←0.
12892          000052                  MICPC=MICPC+1
12893  032070  010000                  .WORD   .$.
12894
12895  032072                  2$:     MOVE    # 0,BREG                  ;
12896          000053                  MICPC=MICPC+1
12897  032072  000400                  .WORD   .$.
12898  032074                          SADD    SPAD <1>,BREG    ;CLEAR C BIT.
12899          000054                  MICPC=MICPC+1
12900  032074  060401                  .WORD   .$.!.DD
12901  032076                          MOVE    MEM,SPAD <0> MARINC      ;GET THE FIRST OPERAND.
12902          000055                  MICPC=MICPC+1
12903  032076  057220                  .WORD   .$.
12904  032100                          SADC    SPAD <0>,BR.SP,MARINC    ;
12905          000056                  MICPC=MICPC+1
12906  032100  077500                  .WORD   .$.
12907  032102                          SIFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
12908
12909
12910  032102                          SUB2C   SPAD <0>,MEM,NOP
12911          000057                  MICPC=MICPC+1
12912  032102  040360                  .WORD   .$.
12913  032104                          BZ      3$
12914          000060                  MICPC=MICPC+1
12915  032104  101473                  .WORD   .$.
12916  032106                          MOVE    MEM,OUT1 <4>     ;GOOD DATA
12917          000061                  MICPC=MICPC+1
12918  032106  041224                  .WORD   .$.
12919  032110                          MOVE    BREG,OUT1 <5>    ;BAD DATA.
12920          000062                  MICPC=MICPC+1
12921  032110  061225                  .WORD   .$.
12922  032112                          MOVE    # 15,BREG        ;SET TYPE OF ERROR.
12923          000063                  MICPC=MICPC+1
12924  032112  000415                  .WORD   .$.
12925  032114                          MOVE    BREG,OUT1 <3>    ;SET TYPE OF ERROR.
12926          000064                  MICPC=MICPC+1
12927  032114  061223                  .WORD   .$.
12928  032116                          MOVE    # 04,BREG        ;LOAD FUNCTION CODE...
12929          000065                  MICPC=MICPC+1
12930  032116  000404                  .WORD   .$.
12931  032120                          MOVE    BREG,OUT1 <CSR7>         ;LOAD IT...
12932          000066                  MICPC=MICPC+1
12933  032120  061227                  .WORD   .$.
12934  032122                          CALL    ERROR            ;ALU A PLUS C ERROR...
12935  032122                          MOVE    # <MICPC+3>,BREG
12936          000067                  MICPC=MICPC+1
12937  032122  000471                  .WORD   .$.
12938  032124                          SBR     ERROR
12939          000070                  MICPC=MICPC+1
```

```
12940  032124  104400                  .WORD   .$.
12941  032126                          MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
12942          000071                  MICPC=MICPC+1
12943  032126  070204                  .WORD   .$.
12944  032130                          SBR     2$              ;LOOP ON ERROR...
12945          000072                  MICPC=MICPC+1
12946  032130  100453                  .WORD   .$.
12947  032132                  3$:     CALL    SCP1
12948  032132                          MOVE    # <MICPC+3>,BREG
12949          000073                  MICPC=MICPC+1
12950  032132  000475                  .WORD   .$.
12951  032134                          SBR     SCP1
12952          000074                  MICPC=MICPC+1
12953  032134  104427                  .WORD   .$.
12954  032136                          MOVE    SPAD <4>,MLR    ;
12955          000075                  MICPC=MICPC+1
12956  032136  070204                  .WORD   .$.
12957  032140                          SBR     2$              ; SCOPE THE DATA....
12958          000076                  MICPC=MICPC+1
12959  032140  100453                  .WORD   .$.
12960  032142                  6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
12961          000077                  MICPC=MICPC+1
12962  032142  070204                  .WORD   .$.
12963  032144                          MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
12964          000100                  MICPC=MICPC+1
12965  032144  057220                  .WORD   .$.
12966  032146                          MOVE    # 377,BREG      ;
12967          000101                  MICPC=MICPC+1
12968  032146  000777                  .WORD   .$.
12969  032150                          SADD    SPAD <2>,BREG   ;SET C BIT...
12970          000102                  MICPC=MICPC+1
12971  032150  060402                  .WORD   .$.!.DD
12972  032152                          SADC    SPAD <0>,BR.SP,MARINC           ;SP 0 & BR = A PLUS C
12973          000103                  MICPC=MICPC+1
12974  032152  077500                  .WORD   .$.
12975  032154                          MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
12976          000104                  MICPC=MICPC+1
12977  032154  057223                  .WORD   .$.
12978  032156                          SIFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
12979
12980
12981  032156                          SUB2C   SPAD <0>,MEM,NOP
12982          000105                  MICPC=MICPC+1
12983  032156  040360                  .WORD   .$.
12984  032160                          BZ      9$
12985          000106                  MICPC=MICPC+1
12986  032160  101521                  .WORD   .$.
12987  032162                          MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
12988          000107                  MICPC=MICPC+1
12989  032162  041224                  .WORD   .$.
12990  032164                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
12991          000110                  MICPC=MICPC+1
12992  032164  061225                  .WORD   .$.
12993  032166                          MOVE    # 23,BREG               ;ERROR TYPE...
12994          000111                  MICPC=MICPC+1
12995  032166  000423                  .WORD   .$.
```

```
12996  032170
12997                 000112
12998  032170  061223
12999  032172
13000                 000113
13001  032172  000404
13002  032174
13003                 000114
13004  032174  061227
13005  032176
13006  032176
13007                 000115
13008  032176  000517
13009  032200
13010                 000116
13011  032200  104400
13012  032202
13013                 000117
13014  032202  070204
13015  032204
13016                 000120
13017  032204  100477
13018  032206                    9$:   CALL    SCP1            ;SCOPE THE ERROR...
13019  032206                          MOVE    # <MICPC+3>,BREG
13020                 000121           MICPC=MICPC+1
13021  032206  000523                 .WORD   .$.
13022  032210                          SBR     SCP1
13023                 000122           MICPC=MICPC+1
13024  032210  104427                 .WORD   .$.
13025  032212                          MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
13026                 000123           MICPC=MICPC+1
13027  032212  070204                 .WORD   .$.
13028  032214                          SBR     6$              ;SCOPE THE DATA...
13029                 000124           MICPC=MICPC+1
13030  032214  100477                 .WORD   .$.
13031  032216                          MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
13032                 000125           MICPC=MICPC+1
13033  032216  000404                 .WORD   .$.
13034  032220                          SADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
13035  032220  077004                 .WORD   .$.!MARINC!.DO
13036  032222                          SDEC    SPAD <7>                ;IS IT DONE??
13037                 000126           MICPC=MICPC+1
13038  032222  063167          .WORD   .$.!.DSP
13039  032224                          BZ      4$                      ;YES, SCOPE THE TEST.
13040                 000127           MICPC=MICPC+1
13041  032224  101532                 .WORD   .$.
13042  032226                          SBR     2$                      ;DO, THE NEXT.
13043                 000130           MICPC=MICPC+1
13044  032226  100453                 .WORD   .$.
13045  032230                    4$:   CALL    SCPE            ;SCOPE THE TEST...
13046  032230                          MOVE    # <MICPC+3>,BREG
13047                 000131           MICPC=MICPC+1
13048  032230  000533                 .WORD   .$.
13049  032232                          SBR     SCPE
13050                 000132           MICPC=MICPC+1
13051  032232  104454                 .WORD   .$.
```

```
13052 032234                          SBR     1$              ;DO THE NEXT ITERATION...
13053         000133                  MICPC=MICPC+1
13054 032234  100400                  .WORD   .$.
13055 032236                  $ALUT1  0,<2'S COMP SUB>,SUB2C,-1,-1,376,376,0,0,-1,-1,-1,-1,124,124,252,252,-1,-1,<A-B-
13056 032236                  $XZ
13057
13058
13059                                 ;*************************** TEST 60 ****************************
13060                                 ;*ALU TEST
13061                                 ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT CLEARED.
13062                                 ;*TEST OF ALU FUNCTION 2'S COMP SUB WITH C BIT SET.
13063                                 ;*ALU FUNCTION (A-B-1)
13064                                 ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
13065                                 ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
13066 032236                  $XZ
13067                                 ;:*************************************************************************
13068
13069 032236                  $TSTN
13070                                 ;  TEST 60
13071                                 ;-------------
13072 032236  012737  000060  001202 TST60:  MOV     #60,$TSTNM              ; LOAD THE NO. OF THIS TEST
13073 032244  012737  032564  001442         MOV     #TST61,NEXT             ; POINT TO THE START OF NEXT TEST.
13074                                                               ;R1 CONTAINS BASE KMC11 ADDRESS
13075 032252  004737  035536                 JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
13076 032256  032272                         MCT60
13077 032260  104022                         ERROR   22                      ;TIME OUT ERROR...
13078 032262  012706  001200                 MOV     #STACK,SP               ;RESET STACK...
13079 032266  000177  147150                 JMP     @NEXT                   ;GO TO NEXT TEST...
13080 032272                  MCT60:
13081 032272                  1$:       MOVE    # 0,MLR          ;SET MAR+LO.
13082         000000                    MICPC=MICPC+1
13083 032272  010000                    .WORD   .$.
13084 032274                            MOVE    # 0,MPR          ;SET MAR+HI.
13085         000001                    MICPC=MICPC+1
13086 032274  004000                    .WORD   .$.
13087 032276                            MOVE    # 0,BREG                 ;
13088         000002                    MICPC=MICPC+1
13089 032276  000400                    .WORD   .$.
13090 032300                            MOVE    BREG,SPAD (16)          ;FOR RETURN ADDRESS...
13091         000003                    MICPC=MICPC+1
13092 032300  063236                    .WORD   .$.
13093 032302                            MOVE    BREG,SPAD (0)   ;
13094         000004                    MICPC=MICPC+1
13095 032302  063220                    .WORD   .$.
13096 032304                            MOVE    BREG,SPAD (1)   ;
13097         000005                    MICPC=MICPC+1
13098 032304  063221                    .WORD   .$.
13099 032306                            MOVE    BREG,SPAD (2)   ;
13100         000006                    MICPC=MICPC+1
13101 032306  063222                    .WORD   .$.
13102 032310                            SDEC    SPAD (2)                ;
13103         000007                    MICPC=MICPC+1
13104 032310  063162          .WORD     .$.!.DSP
13105 032312                            MOVE    BREG,SPAD (4)           ;
13106         000010                    MICPC=MICPC+1
13107 032312  063224                    .WORD   .$.
```

```
13108  032314                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
13109          000011           MICPC=MICPC+1
13110  032314  016400           .WORD   .S.
13111  032316                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
13112          000012           MICPC=MICPC+1
13113  032316  016400           .WORD   .S.
13114  032320                  MOVE    # -1,MEM MARINC
13115          000013           MICPC=MICPC+1
13116  032320  016777           .WORD   .S.
13117  032322                  MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
13118          000014           MICPC=MICPC+1
13119  032322  016777           .WORD   .S.
13120  032324                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
13121          000015           MICPC=MICPC+1
13122  032324  016777           .WORD   .S.
13123  032326                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
13124          000016           MICPC=MICPC+1
13125  032326  016400           .WORD   .S.
13126  032330                  MOVE    # 376,MEM MARINC
13127          000017           MICPC=MICPC+1
13128  032330  016776           .WORD   .S.
13129  032332                  MOVE    # 376,MEM MARINC        ;RESULT WITH C BIT SET.
13130          000020           MICPC=MICPC+1
13131  032332  016776           .WORD   .S.
13132  032334                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
13133          000021           MICPC=MICPC+1
13134  032334  016400           .WORD   .S.
13135  032336                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
13136          000022           MICPC=MICPC+1
13137  032336  016777           .WORD   .S.
13138  032340                  MOVE    # 0,MEM MARINC
13139          000023           MICPC=MICPC+1
13140  032340  016400           .WORD   .S.
13141  032342                  MOVE    # 0,MEM MARINC  ;RESULT WITH C BIT SET.
13142          000024           MICPC=MICPC+1
13143  032342  016400           .WORD   .S.
13144  032344                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY
13145          000025           MICPC=MICPC+1
13146  032344  016777           .WORD   .S.
13147  032346                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
13148          000026           MICPC=MICPC+1
13149  032346  016777           .WORD   .S.
13150  032350                  MOVE    # -1,MEM MARINC
13151          000027           MICPC=MICPC+1
13152  032350  016777           .WORD   .S.
13153  032352                  MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
13154          000030           MICPC=MICPC+1
13155  032352  016777           .WORD   .S.
13156  032354                  MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
13157          000031           MICPC=MICPC+1
13158  032354  016525           .WORD   .S.
13159  032356                  MOVE    # 125,MEM MARINC        ;LOAD THE DATA IN MEMORY.
13160          000032           MICPC=MICPC+1
13161  032356  016525           .WORD   .S.
13162  032360                  MOVE    # -1,MEM MARINC
13163          000033           MICPC=MICPC+1
```

```
13164  032360  016777          .WORD   .S.
13165  032362                  MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
13166          000034          MICPC=MICPC+1
13167  032362  016777          .WORD   .S.
13168  032364                  MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13169          000035          MICPC=MICPC+1
13170  032364  016652          .WORD   .S.
13171  032366                  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13172          000036          MICPC=MICPC+1
13173  032366  016525          .WORD   .S.
13174  032370                  MOVE    # 124,MEM MARINC
13175          000037          MICPC=MICPC+1
13176  032370  016524          .WORD   .S.
13177  032372                  MOVE    # 124,MEM MARINC         ;RESULT WITH C BIT SET.
13178          000040          MICPC=MICPC+1
13179  032372  016524          .WORD   .S.
13180  032374                  MOVE    # 125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13181          000041          MICPC=MICPC+1
13182  032374  016525          .WORD   .S.
13183  032376                  MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY
13184          000042          MICPC=MICPC+1
13185  032376  016652          .WORD   .S.
13186  032400                  MOVE    # 252,MEM MARINC
13187          000043          MICPC=MICPC+1
13188  032400  016652          .WORD   .S.
13189  032402                  MOVE    # 252,MEM MARINC         ;RESULT WITH C BIT SET.
13190          000044          MICPC=MICPC+1
13191  032402  016652          .WORD   .S.
13192  032404                  MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13193          000045          MICPC=MICPC+1
13194  032404  016652          .WORD   .S.
13195  032406                  MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13196          000046          MICPC=MICPC+1
13197  032406  016652          .WORD   .S.
13198  032410                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
13199          000047          MICPC=MICPC+1
13200  032410  016777          .WORD   .S.
13201  032412                  MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
13202          000050          MICPC=MICPC+1
13203  032412  016777          .WORD   .S.
13204  032414                  MOVE    # 7,SPAD <7>     ;SET THE COUNT.
13205          000051          MICPC=MICPC+1
13206  032414  003007          .WORD   .S.
13207
13208  032416                  MOVE    # 0,MLR          ;MAR←0.
13209          000052          MICPC=MICPC+1
13210  032416  010000          .WORD   .S.
13211
13212  032420                  MOVE    # 0,BREG                 ;
13213          000053          MICPC=MICPC+1
13214  032420  000400          .WORD   .S.
13215  032422                  SADD    SPAD <1>,BREG   ;CLEAR C BIT.
13216          000054          MICPC=MICPC+1
13217  032422  060401          .WORD   .S.!.DD
13218  032424                  MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
13219          000055          MICPC=MICPC+1
```

```
13220  032424  057220                  .WORD   .$.
13221  032426                          SUB2C   MEM,SPAD <0>,BR.SP,MARINC      ;SPAD  <0>:=$FUNC.
13222          000056                  MICPC=MICPC+1
13223  032426  057760                  .WORD   .$.
13224  032430                          $IFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
13225
13226
13227  032430                          SUB2C   SPAD <0>,MEM,NOP
13228          000057                  MICPC=MICPC+1
13229  032430  040360                  .WORD   .$.
13230  032432                          BZ      3$
13231          000060                  MICPC=MICPC+1
13232  032432  101473                  .WORD   .$.
13233  032434                          MOVE    MEM,OUT1 <4>    ;GOOD DATA
13234          000061                  MICPC=MICPC+1
13235  032434  041224                  .WORD   .$.
13236  032436                          MOVE    BREG,OUT1 <5>   ;BAD DATA.
13237          000062                  MICPC=MICPC+1
13238  032436  061225                  .WORD   .$.
13239  032440                          MOVE    # 15,BREG       ;SET TYPE OF ERROR.
13240          000063                  MICPC=MICPC+1
13241  032440  000415                  .WORD   .$.
13242  032442                          MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
13243          000064                  MICPC=MICPC+1
13244  032442  061223                  .WORD   .$.
13245  032444                          MOVE    # 17,BREG       ;LOAD FUNCTION CODE...
13246          000065                  MICPC=MICPC+1
13247  032444  000417                  .WORD   .$.
13248  032446                          MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
13249          000066                  MICPC=MICPC+1
13250  032446  061227                  .WORD   .$.
13251  032450                          CALL    EROR            ;ALU 2'S COMP SUB ERROR...
13252  032450                          MOVE    # <MICPC+3>,BREG
13253          000067                  MICPC=MICPC+1
13254  032450  000471                  .WORD   .$.
13255  032452                          SBR     EROR
13256          000070                  MICPC=MICPC+1
13257  032452  104400                  .WORD   .$.
13258  032454                          MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
13259          000071                  MICPC=MICPC+1
13260  032454  070204                  .WORD   .$.
13261  032456                          SBR     2$              ;LOOP ON ERROR...
13262          000072                  MICPC=MICPC+1
13263  032456  100453                  .WORD   .$.
13264  032460                  3$:     CALL    SCP1
13265  032460                          MOVE    # <MICPC+3>,BREG
13266          000073                  MICPC=MICPC+1
13267  032460  000475                  .WORD   .$.
13268  032462                          SBR     SCP1
13269          000074                  MICPC=MICPC+1
13270  032462  104427                  .WORD   .$.
13271  032464                          MOVE    SPAD <4>,MLR    ;
13272          000075                  MICPC=MICPC+1
13273  032464  070204                  .WORD   .$.
13274  032466                          SBR     2$              ; SCOPE THE DATA....
13275          000076                  MICPC=MICPC+1
```

```
13276  032466  100453                    .WORD   .$
13277  032470                    6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
13278          000077                    MICPC=MICPC+1
13279  032470  070204                    .WORD   .$.
13280  032472                            MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
13281          000100                    MICPC=MICPC+1
13282  032472  057220                    .WORD   .$.
13283  032474                            MOVE    # 377,BREG      ;
13284          000101                    MICPC=MICPC+1
13285  032474  000777                    .WORD   .$.
13286  032476                            SADD    SPAD <2>,BREG   ;SET C BIT...
13287          000102                    MICPC=MICPC+1
13288  032476  060402                    .WORD   .$.!.DD
13289  032500                            SUB2C   MEM,SPAD <0>,BR.SP,MARINC       ;SP 0 & BR = 2'S COMP SUB
13290          000103                    MICPC=MICPC+1
13291  032500  057760                    .WORD   .$.
13292  032502                            MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
13293          000104                    MICPC=MICPC+1
13294  032502  057223                    .WORD   .$.
13295  032504                            $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
13296
13297
13298  032504                            SUB2C   SPAD <0>,MEM,NOP
13299          000105                    MICPC=MICPC+1
13300  032504  040360                    .WORD   .$.
13301  032506                            BZ      &$
13302          000106                    MICPC=MICPC+1
13303  032506  101521                    .WORD   .$.
13304  032510                            MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
13305          000107                    MICPC=MICPC+1
13306  032510  041224                    .WORD   .$.
13307  032512                            MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
13308          000110                    MICPC=MICPC+1
13309  032512  061225                    .WORD   .$.
13310  032514                            MOVE    # 23,BREG               ;ERROR TYPE...
13311          000111                    MICPC=MICPC+1
13312  032514  000423                    .WORD   .$.
13313  032516                            MOVE    BREG,OUT1 <CSR3>        ;
13314          000112                    MICPC=MICPC+1
13315  032516  061223                    .WORD   .$.
13316  032520                            MOVE    # 17,BREG               ;ALU FUNCTION CODE.
13317          000113                    MICPC=MICPC+1
13318  032520  000417                    .WORD   .$.
13319  032522                            MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
13320          000114                    MICPC=MICPC+1
13321  032522  061227                    .WORD   .$.
13322  032524                            CALL    ERROR   ;REPORT ERROR...
13323  032524                            MOVE    # <MICPC+3>,BREG
13324          000115                    MICPC=MICPC+1
13325  032524  000517                    .WORD   .$.
13326  032526                            SBR     ERROR
13327          000116                    MICPC=MICPC+1
13328  032526  104400                    .WORD   .$.
13329  032530                            MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
13330          000117                    MICPC=MICPC+1
13331  032530  070204                    .WORD   .$.
```

```
13332 032532                          SBR     6$                      ;LOOP ON ERROR...
13333 000120                          MICPC=MICPC+1
13334 032532 100477                   .WORD   .$.
13335 032534                    9$:   CALL    SCP1                    ;SCOPE THE ERROR...
13336 032534                          MOVE    @ <MICPC+3>,BREG
13337 000121                          MICPC=MICPC+1
13338 032534 000523                   .WORD   .$.
13339 032536                          SBR     SCP1
13340 000122                          MICPC=MICPC+1
13341 032536 104427                   .WORD   .$.
13342 032540                          MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
13343 000123                          MICPC=MICPC+1
13344 032540 070204                   .WORD   .$.
13345 032542                          SBR     6$              ;SCOPE THE DATA...
13346 000124                          MICPC=MICPC+1
13347 032542 100477                   .WORD   .$.
13348 032544                          MOVE    @ 4,BREG                ;UPDATE BACKGROUND POINTER.
13349 000125                          MICPC=MICPC+1
13350 032544 000404                   .WORD   .$.
13351 032546                          $ADD    BREG,SPAD <4>,MARINC    ;ALSO DATA POINTER.
13352 032546 077004                   .WORD   .$.!MARINC!.DO
13353 032550                          $DEC    SPAD <7>                ;IS IT DONE??
13354 000126                          MICPC=MICPC+1
13355 032550 063167             .WORD .$.!.DSP
13356 032552                          BZ      4$              ;YES, SCOPE THE TEST.
13357 000127                          MICPC=MICPC+1
13358 032552 101532                   .WORD   .$.
13359 032554                          SBR     2$              ;DO, THE NEXT.
13360 000130                          MICPC=MICPC+1
13361 032554 100453                   .WORD   .$.
13362 032556                    4$:   CALL    $CPE            ;SCOPE THE TEST...
13363 032556                          MOVE    @ <MICPC+3>,BREG
13364 000131                          MICPC=MICPC+1
13365 032556 000533                   .WORD   .$.
13366 032560                          SBR     $CPE
13367 000132                          MICPC=MICPC+1
13368 032560 104454                   .WORD   .$.
13369 032562                          SBR     1$              ;DO THE NEXT ITERATION...
13370 000133                          MICPC=MICPC+1
13371 032562 100400                   .WORD   .$.
13372 032564                    $ALUT1 0,<DEC A>,$DEC,-1,-1,376,376,-1,-1,376,376,124,124,251,251,124,124,251,251,<A-1>
13373 032564                    $XZ
13374
13375
13376                                 ;*************************** TEST 61 ****************************
13377                                 ;*ALU TEST
13378                                 ;*TEST OF ALU FUNCTION DEC A WITH C BIT CLEARED.
13379                                 ;*TEST OF ALU FUNCTION DEC A WITH C BIT SET.
13380                                 ;*ALU FUNCTION (A-1)
13381                                 ;*LOAD MAIN MEMORY 16 WORDS OF DATA.
13382                                 ;*PERFORM THE FUNCTION, VERIFY THE RESULTS..
13383 032564                    $XZ
13384                                 ;;************************************************************************
13385
13386 032564                    $TSTN
13387                                 ;  TEST 61
```

```
                                        ;--------------
032564 012737 000061 001202 TST61:      MOV     #61,$TSTNM              ; LOAD THE NO. OF THIS TEST
032572 012737 033112 001442             MOV     #TST62,NEXT             ; POINT TO THE START OF NEXT TEST.
                                    ;R1 CONTAINS BASE KMC11 ADDRESS
032600 004737 035536                    JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
032604 032620                           MCT61
032606 104022                           ERROR   22                      ;TIME OUT ERROR...
032610 012706 001200                    MOV     #STACK,SP               ;RESET STACK...
032614 000177 146622                    JMP     @NEXT                   ;GO TO NEXT TEST...
032620                          MCT61:
032620                          1$:     MOVE    # 0,MLR          ;SET MAR←LO.
000000                                  MICPC=MICPC+1
032620 010000                           .WORD   .$.
032622                                  MOVE    # 0,MPR          ;SET MAR←HI.
000001                                  MICPC=MICPC+1
032622 004000                           .WORD   .$.
032624                                  MOVE    # 0,BREG                 ;
000002                                  MICPC=MICPC+1
032624 000400                           .WORD   .$.
032626                                  MOVE    BREG,SPAD <16>          ;FOR RETURN ADDRESS...
000003                                  MICPC=MICPC+1
032626 063236                           .WORD   .$.
032630                                  MOVE    BREG,SPAD <0>   ;
000004                                  MICPC=MICPC+1
032630 063220                           .WORD   .$.
032632                                  MOVE    BREG,SPAD <1>   ;
000005                                  MICPC=MICPC+1
032632 063221                           .WORD   .$.
032634                                  MOVE    BREG,SPAD <2>   ;
000006                                  MICPC=MICPC+1
032634 063222                           .WORD   .$.
032636                                  SDEC    SPAD <2>                ;
000007                                  MICPC=MICPC+1
032636 063162                   .WORD   .$.!.DSP
032640                                  MOVE    BREG,SPAD <4>           ;
000010                                  MICPC=MICPC+1
032640 063224                           .WORD   .$.
032642                                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
000011                                  MICPC=MICPC+1
032642 016400                           .WORD   .$.
032644                                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY
000012                                  MICPC=MICPC+1
032644 016400                           .WORD   .$.
032646                                  MOVE    # -1,MEM MARINC
000013                                  MICPC=MICPC+1
032646 016777                           .WORD   .$.
032650                                  MOVE    # -1,MEM MARINC ;RESULT WITH C BIT SET.
000014                                  MICPC=MICPC+1
032650 016777                           .WORD   .$.
032652                                  MOVE    # -1,MEM MARINC ;LOAD THE DATA IN MEMORY.
000015                                  MICPC=MICPC+1
032652 016777                           .WORD   .$.
032654                                  MOVE    # 0,MEM MARINC  ;LOAD THE DATA IN MEMORY.
000016                                  MICPC=MICPC+1
032654 016400                           .WORD   .$.
032656                                  MOVE    # 376,MEM MARINC
```

```
13444                   000017          MICPC=MICPC+1
13445 032656 016776                     .WORD   .S.
13446 032660                            MOVE    #376,MEM MARINC         ;RESULT WITH C BIT SET.
13447                   000020          MICPC=MICPC+1
13448 032660 016776                     .WORD   .S.
13449 032662                            MOVE    #0,MEM MARINC   ;LOAD THE DATA IN MEMORY.
13450                   000021          MICPC=MICPC+1
13451 032662 016400                     .WORD   .S.
13452 032664                            MOVE    #-1,MEM MARINC ;LOAD THE DATA IN MEMORY.
13453                   000022          MICPC=MICPC+1
13454 032664 016777                     .WORD   .S.
13455 032666                            MOVE    #-1,MEM MARINC
13456                   000023          MICPC=MICPC+1
13457 032666 016777                     .WORD   .S.
13460 032670                            MOVE    #-1,MEM MARINC ;RESULT WITH C BIT SET.
13459                   000024          MICPC=MICPC+1
13460 032670 016777                     .WORD   .S.
13461 032672                            MOVE    #-1,MEM MARINC ;LOAD THE DATA IN MEMORY
13462                   000025          MICPC=MICPC+1
13463 032672 016777                     .WORD   .S.
13464 032674                            MOVE    #-1,MEM MARINC ;LOAD THE DATA IN MEMORY.
13465                   000026          MICPC=MICPC+1
13466 032674 016777                     .WORD   .S.
13467 032676                            MOVE    #376,MEM MARINC
13470                   000027          MICPC=MICPC+1
13471 032676 016776                     .WORD   .S.
13472 032700                            MOVE    #376,MEM MARINC         ;RESULT WITH C BIT SET.
13473                   000030          MICPC=MICPC+1
13474 032700 016776                     .WORD   .S.
13475 032702                            MOVE    #125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13476                   000031          MICPC=MICPC+1
13477 032702 016525                     .WORD   .S.
13500 032704                            MOVE    #125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13501                   000032          MICPC=MICPC+1
13502 032704 016525                     .WORD   .S.
13503 032706                            MOVE    #124,MEM MARINC
13504                   000033          MICPC=MICPC+1
13505 032706 016524                     .WORD   .S.
13506 032710                            MOVE    #124,MEM MARINC         ;RESULT WITH C BIT SET.
13507                   000034          MICPC=MICPC+1
13510 032710 016524                     .WORD   .S.
13511 032712                            MOVE    #252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13512                   000035          MICPC=MICPC+1
13513 032712 016652                     .WORD   .S.
13514 032714                            MOVE    #125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13515                   000036          MICPC=MICPC+1
13516 032714 016525                     .WORD   .S.
13517 032716                            MOVE    #251,MEM MARINC
13520                   000037          MICPC=MICPC+1
13521 032716 016651                     .WORD   .S.
13522 032720                            MOVE    #251,MEM MARINC         ;RESULT WITH C BIT SET.
13523                   000040          MICPC=MICPC+1
13524 032720 016651                     .WORD   .S.
13525 032722                            MOVE    #125,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13526                   000041          MICPC=MICPC+1
13527 032722 016525                     .WORD   .S.
```

```
13500 032724                    MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY
13501         000042            MICPC=MICPC+1
13502 032724  016652            .WORD   .S.
13503 032726                    MOVE    # 124,MEM MARINC
13504         000043            MICPC=MICPC+1
13505 032726  016524            .WORD   .S.
13506 032730                    MOVE    # 124,MEM MARINC         ;RESULT WITH C BIT SET.
13507         000044            MICPC=MICPC+1
13508 032730  016524            .WORD   .S.
13509 032732                    MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13510         000045            MICPC=MICPC+1
13511 032732  016652            .WORD   .S.
13512 032734                    MOVE    # 252,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13513         000046            MICPC=MICPC+1
13514 032734  016652            .WORD   .S.
13515 032736                    MOVE    # 251,MEM MARINC         ;LOAD THE DATA IN MEMORY.
13516         000047            MICPC=MICPC+1
13517 032736  016651            .WORD   .S.
13518 032740                    MOVE    # 251,MEM MARINC         ;RESULT WITH C BIT SET.
13519         000050            MICPC=MICPC+1
13520 032740  016651            .WORD   .S.
13521 032742                    MOVE    # 7,SPAD <7>     ;SET THE COUNT.
13522         000051            MICPC=MICPC+1
13523 032742  003007            .WORD   .S.
13524
13525 032744                    MOVE    # 0,MLR          ;MAR=0.
13526         000052            MICPC=MICPC+1
13527 032744  010000            .WORD   .S.
13528
13529 032746                25: MOVE    # 0,BREG                  ;
13530         000053            MICPC=MICPC+1
13531 032746  000400            .WORD   .S.
13532 032750                    $AOD    SPAD <1>,BREG   ;CLEAR C BIT.
13533         000054            MICPC=MICPC+1
13534 032750  060401            .WORD   .S.!.DD
13535 032752                    MOVE    MEM,SPAD <0> MARINC     ;GET THE FIRST OPERAND.
13536         000055            MICPC=MICPC+1
13537 032752  057220            .WORD   .S.
13538 032754                    $DEC    SPAD <0>,BR.SP,MARINC   ;
13539         000056            MICPC=MICPC+1
13540 032754  077560            .WORD   .S.
13541 032756                    SIFEQ   MEM,SPAD <0> 3$ ;BRANCH IF GOOD.
13542
13543
13544 032756                    SUB2C   SPAD <0>,MEM,NOP
13545         000057            MICPC=MICPC+1
13546 032756  040360            .WORD   .S.
13547 032760                    BZ      3$
13548         000060            MICPC=MICPC+1
13549 032760  101473            .WORD   .S.
13550 032762                    MOVE    MEM,OUT1 <4>    ;GOOD DATA
13551         000061            MICPC=MICPC+1
13552 032762  041224            .WORD   .S.
13553 032764                    MOVE    BREG,OUT1 <5>   ;BAD DATA.
13554         000062            MICPC=MICPC+1
13555 032764  061225            .WORD   .S.
```

```
13556 032766                    MOVE    # 15,BREG       ;SET TYPE OF ERROR.
13557       000063              MICPC=MICPC+1
13560 032766 000415             .WORD   .$.
13561 032770                    MOVE    BREG,OUT1 <3>   ;SET TYPE OF ERROR.
13562       000064              MICPC=MICPC+1
13563 032770 061223             .WORD   .$.
13564 032772                    MOVE    # 07,BREG       ;LOAD FUNCTION CODE...
13565       000065              MICPC=MICPC+1
13566 032772 000407             .WORD   .$.
13567 032774                    MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
13570       000066              MICPC=MICPC+1
13571 032774 061227             .WORD   .$.
13572 032776                    CALL    ERROR           ;ALU DEC A ERROR...
13573       000067              MOVE    # <MICPC+3>,BREG
13574 032776 000471             MICPC=MICPC+1
13575 033000                    .WORD   .$.
13576       000070              SBR     ERROR
13577 033000 104400             MICPC=MICPC+1
13600 033002                    .WORD   .$.
13601       000071              MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
13602 033002 070204             MICPC=MICPC+1
13603 033004                    .WORD   .$.
13604       000072              SBR     2$              ;LOOP ON ERROR...
13605 033004 100453             MICPC=MICPC+1
13606 033006                    .WORD   .$.
13607       000073      3$:     CALL    SCP1
13610 033006 000475             MOVE    # <MICPC+3>,BREG
13611 033010                    MICPC=MICPC+1
13612       000074              .WORD   .$.
13613 033010 104427             SBR     SCP1
13614 033012                    MICPC=MICPC+1
13615       000075              .WORD   .$.
13616 033012 070204             MOVE    SPAD <4>,MLR    ;
13617 033014                    MICPC=MICPC+1
13620       000076              .WORD   .$.
13621 033014 100453             SBR     2$              ; SCOPE THE DATA....
13622 033016                    MICPC=MICPC+1
13623       000077              .WORD   .$.
13624 033016 070204     6$:     MOVE    SPAD <4>,MLR    ;RESET DATA POINTER...
13625 033020                    MICPC=MICPC+1
13626       000100              .WORD   .$.
13627 033020 057220             MOVE    MEM,SPAD <0>,MARINC     ;GET FIRST OPRAND...
13630 033022                    MICPC=MICPC+1
13631       000101              .WORD   .$.
13632 033022 000777             MOVE    # 377,BREG      ;
13633 033024                    MICPC=MICPC+1
13634       000102              .WORD   .$.
13635 033024 060402             $ADD    SPAD <2>,BREG   ;SET C BIT...
13636 033026                    MICPC=MICPC+1
13637       000103              .WORD   .$.!.DD
13640 033026 077560             $DEC    SPAD <0>,BR.SP,MARINC           ;SP 0 & BR = DEC A
13641 033030                    MICPC=MICPC+1
13642       000104              .WORD   .$.
13643 033030 057223             MOVE    MEM,SPAD <3>,MARINC     ;DUMMY INSTR, TO MARINC.
                                MICPC=MICPC+1
                                .WORD   .$.
```

```
13612  033032                          $IFEQ   MEM,SPAD <0>    9$      ;BR IF GOOD...
13613
13614
13615  033032                          SUB2C   SPAD <0>,MEM,NOP
13616          000105                  MICPC=MICPC+1
13617  033032  040360                  .WORD   .$.
13618  033034                          BZ      9$
13619          000106                  MICPC=MICPC+1
13620  033034  101521                  .WORD   .$.
13621  033036                          MOVE    MEM,OUT1 <CSR4> ;GOOD DATA.
13622          000107                  MICPC=MICPC+1
13623  033036  041224                  .WORD   .$.
13624  033040                          MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
13625          000110                  MICPC=MICPC+1
13626  033040  061225                  .WORD   .$.
13627  033042                          MOVE    # 23,BREG               ;ERROR TYPE...
13628          000111                  MICPC=MICPC+1
13629  033042  000423                  .WORD   .$.
13630  033044                          MOVE    BREG,OUT1 <CSR3>        ;
13631          000112                  MICPC=MICPC+1
13632  033044  061223                  .WORD   .$.
13633  033046                          MOVE    # 07,BREG               ;ALU FUNCTION CODE.
13634          000113                  MICPC=MICPC+1
13635  033046  000407                  .WORD   .$.
13636  033050                          MOVE    BREG,OUT1 <CSR7>        ;LOAD IT...
13637          000114                  MICPC=MICPC+1
13638  033050  061227                  .WORD   .$.
13639  033052                          CALL    ERROR   ;REPORT ERROR...
13640  033052                          MOVE    # <MICPC+3>,BREG
13641          000115                  MICPC=MICPC+1
13642  033052  000517                  .WORD   .$.
13643  033054                          SBR     ERROR
13644          000116                  MICPC=MICPC+1
13645  033054  104400                  .WORD   .$.
13646  033056                          MOVE    SPAD <4>,MLR            ;RESTORE DATA POINTER.
13647          000117                  MICPC=MICPC+1
13648  033056  070204                  .WORD   .$.
13649  033060                          SBR     6$                      ;LOOP ON ERROR...
13650          000120                  MICPC=MICPC+1
13651  033060  100477                  .WORD   .$.
13652  033062                  9$:     CALL    SCP1            ;SCOPE THE ERROR...
13653  033062                          MOVE    # <MICPC+3>,BREG
13654          000121                  MICPC=MICPC+1
13655  033062  000523                  .WORD   .$.
13656  033064                          SBR     SCP1
13657          000122                  MICPC=MICPC+1
13658  033064  104427                  .WORD   .$.
13659  033066                          MOVE    SPAD <4>,MLR    ;RESTORE DATA POINTER...
13660          000123                  MICPC=MICPC+1
13661  033066  070204                  .WORD   .$.
13662  033070                          SBR     6$              ;SCOPE THE DATA...
13663          000124                  MICPC=MICPC+1
13664  033070  100477                  .WORD   .$.
13665  033072                          MOVE    # 4,BREG                ;UPDATE BACKGROUND POINTER.
13666          000125                  MICPC=MICPC+1
13667  033072  000404                  .WORD   .$.
```

```
13668 033074                        SADD    BREG,SPAD (4),MARINC    ;ALSO DATA POINTER.
13669 033074 077004                 .WORD   .$.!MARINC!.DD
13670 033076                        SDEC    SPAD (7)                ;IS IT DONE??
13671        000126                 MICPC=MICPC+1
13672 033076 063167           .WORD .$.!.DSP
13673 033100                        BZ      4$                      ;YES, SCOPE THE TEST.
13674        000127                 MICPC=MICPC+1
13675 033100 101532                 .WORD   .$.
13676 033102                        SBR     2$                      ;DO, THE NEXT.
13677        000130                 MICPC=MICPC+1
13678 033102 100453                 .WORD   .$.
13679 033104                  4$:   CALL    SCPE                    ;SCOPE THE TEST...
13680 033104                        MOVE    @ (MICPC+3),BREG
13681        000131                 MICPC=MICPC+1
13682 033104 000533                 .WORD   .$.
13683 033106                        SBR     SCPE
13684        000132                 MICPC=MICPC+1
13685 033106 104454                 .WORD   .$.
13686 033110                        SBR     1$                      ;DO THE NEXT ITERATION...
13687        000133                 MICPC=MICPC+1
13688 033110 100400                 .WORD   .$.
13689 033112                  SPGMCK
13690 033112                  SXZ
13691
13692
13693                               ;************************** TEST 62 ****************************
13694                               ;*TEST OF PROGRAM CLOCK BIT
13695                               ;*DO A MASTER CLEAR, VERIFY THAT PROGRAM CLOCK IS SET
13696                               ;*WRITE PROGRAM CLOCK BIT TO A ONE, VERIFY THAT IT CLEARS.
13697                               ;*AND THEN SETS SOME TIME LATER.
13698 033112                  SXZ
13699                               ;:***************************************************************
13700
13701 033112                  STSTN
13702
13703                               ;  TEST 62
13704                               ;-------------
13705 033112 012737 000062 001202 TST62:  MOV     #62,STSTNM              ; LOAD THE NO. OF THIS TEST
13706 033120 012737 033260 001442         MOV     #TST63,NEXT             ; POINT TO THE START OF NEXT TEST.
13707                                                               ;R1 CONTAINS BASE KMC11 ADDRESS
13708 033126 004737 035536                JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
13709 033132 033146                       MCT62
13710 033134 104022                       ERROR   22                      ;TIME OUT ERROR...
13711 033136 012706 001200                MOV     #STACK,SP               ;RESET STACK...
13712 033142 000177 146274                JMP     @NEXT                   ;GO TO NEXT TEST...
13713 033146                        MCT62:
13714 033146                        1$:   MOVE    @ 0,BREG          ;PREPARE
13715        000000                 MICPC=MICPC+1
13716 033146 000400                 .WORD   .$.
13717 033150                        MOVE    BREG,SPAD (2)     ;       FOR
13718        000001                 MICPC=MICPC+1
13719 033150 063222                 .WORD   .$.
13720 033152                        MOVE    BREG,SPAD (3)     ;               DELAY.
13721        000002                 MICPC=MICPC+1
13722 033152 063223                 .WORD   .$.
13723 033154                        MOVE    BREG,SPAD (4)     ;
              000003                MICPC=MICPC+1
```

```
13724 033154 063224                    .WORD   .$.
13725 033156                           MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSE...
13726        000004                    MICPC=MICPC+1
13727 033156 063236                    .WORD   .$.
13728 033160                           MOVE    INP1 <CSR11>,BREG       ;CHECK IF PGM. CLK BIT SET.
13729        000005                    MICPC=MICPC+1
13730 033160 120620                    .WORD   .$.
13731 033162                           BPL     2$                      ;YES, GO RESET IT.
13732        000006                    MICPC=MICPC+1
13733 033162 103014                    .WORD   .$.
13734 033164                           MOVE    #14,MEM                 ;ERROR PROGRAM CLOCK BIT IS NOT SET.
13735        000007                    MICPC=MICPC+1
13736 033164 002414                    .WORD   .$.
13737 033166                           MOVE    MEM,OUT1 <CSR3>         ;ERROR TYPE...
13738        000010                    MICPC=MICPC+1
13739 033166 041223                    .WORD   .$.
13740 033170                           CALL    EROR1                   ;SET TYPE OF ERROR
13741 033170                           MOVE    @ <MICPC+3>,BREG
13742        000011                    MICPC=MICPC+1
13743 033170 000413                    .WORD   .$.
13744 033172                           SBR     EROR1
13745        000012                    MICPC=MICPC+1
13746 033172 104401                    .WORD   .$.
13747 033174                           SBR     1$                      ;LOOP ON ERROR.
13748        000013                    MICPC=MICPC+1
13749 033174 100400                    .WORD   .$.
13750 033176                2$:        MOVE    #20,BREG                ;GET SET TO CLEAR
13751        000014                    MICPC=MICPC+1
13752 033176 000420                    .WORD   .$.
13753 033200                           MOVE    BREG,OUT1 <CSR11>       ;PROGRAM CLOCK BIT.
13754        000015                    MICPC=MICPC+1
13755 033200 061231                    .WORD   .$.
13756 033202                           MOVE    INP1 <CSR11>,BREG       ;SEE IF IT IS CLEARED.
13757        000016                    MICPC=MICPC+1
13758 033202 120620                    .WORD   .$.
13759 033204                           BPL     3$                      ;NO, REPORT ERROR.
13760        000017                    MICPC=MICPC+1
13761 033204 103021                    .WORD   .$.
13762 033206                           SBR     4$                      ;GO SEE IF IT SETS AGAIN.
13763        000020                    MICPC=MICPC+1
13764 033206 100426                    .WORD   .$.
13765 033210                3$:
13766 033210                           MOVE    #14,MEM                 ;TYPE
13767        000021                    MICPC=MICPC+1
13768 033210 002414                    .WORD   .$.
13769 033212                           MOVE    MEM,OUT1 <CSR3>         ;
13770        000022                    MICPC=MICPC+1
13771 033212 041223                    .WORD   .$.
13772 033214                           CALL    EROR1                   ;PGM. CLK BIT DOES NOT CLEAR.
13773 033214                           MOVE    @ <MICPC+3>,BREG
13774        000023                    MICPC=MICPC+1
13775 033214 000425                    .WORD   .$.
13776 033216                           SBR     EROR1
13777        000024                    MICPC=MICPC+1
13778 033216 104401                    .WORD   .$.
13779 033220                           SBR     1$                      ;START AGAIN...
```

```
13780                   000025
13781  033220  100400
13782  033222
13783                   000026
13784  033222  120620
13785  033224
13786                   000027
13787  033224  103042
13788  033226
13789                   000030
13790  033226  063062
13791  033230
13792                   000031
13793  033230  063103
13794  033232
13795                   000032
13796  033232  060603
13797  033234
13798                   000033
13799  033234  103435
13800  033236
13801                   000034
13802  033236  100426
13803  033240
13804  033240
13805                   000035
13806  033240  002414
13807  033242
13808                   000036
13809  033242  041223
13810  033244
13811  033244
13812                   000037
13813  033244  000441
13814  033246
13815                   000040
13816  033246  104401
13817  033250
13818                   000041
13819  033250  100400
13820  033252
13821  033252
13822                   000042
13823  033252  000444
13824  033254
13825                   000043
13826  033254  104454
13827  033256
13828                   000044
13829  033256  100400
13830  033260
13831  033260
13832
13833
13834
13835
```

```
                MICPC=MICPC+1
                .WORD   .$.
4$:             MOVE    INP1 <CSR11>,BREG       ;IS IT SET AGAIN??
                MICPC=MICPC+1
                .WORD   .$.
                BR4     5$                      ;YES, IT'S OK.
                MICPC=MICPC+1
                .WORD   .$.
                $INC    SPAD <2>                ;IN DELAY!
                MICPC=MICPC+1
.WORD           .$.!.DSP
                $AOC    SPAD <3>                ;DELAY!
                MICPC=MICPC+1
.WORD           .$.!.DSP
                MOVE    SPAD <3>,BREG   ;IS DELAYED ENOUGH?
                MICPC=MICPC+1
                .WORD   .$.
                BR7     6$                      ;YES, REPORT ERROR.
                MICPC=MICPC+1
                .WORD   .$.
                $BR     4$                      ;CONTINUE DELAY LOOP.
                MICPC=MICPC+1
                .WORD   .$.
6$:
                MOVE    # 14,MEM                ;SET TYPE OF ERROR
                MICPC=MICPC+1
                .WORD   .$.
                MOVE    MEM,OUT1 <CSR3>         ;
                MICPC=MICPC+1
                .WORD   .$.
                CALL    ERROR1                  ;ERROR PGM. CLK NOT SET.
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     ERROR1
                MICPC=MICPC+1
                .WORD   .$.
                SBR     1$                      ;LOOP ON ERROR.
                MICPC=MICPC+1
                .WORD   .$.
5$:             CALL    SCPE                    ;SCOPE THE TEST.
                MOVE    # <MICPC+3>,BREG
                MICPC=MICPC+1
                .WORD   .$.
                SBR     SCPE
                MICPC=MICPC+1
                .WORD   .$.
                SBR     1$                      ;DO NEXT PASS.
                MICPC=MICPC+1
                .WORD   .$.
$NOISE
$XZ


                ;*************************** TEST 63 ****************************
                ;*MICRO-PROCESSOR NOISE TEST.
```

```
13836                                   ;#WRITE ALL ZERO'S THEN ALL ONE'S THEN A DATA PATTERN TO
13837                                   ;#THE IBUS#,IBUS,SP,S MAIN MEMORY
13838                                   ;#THEN GO AND READ BACK THE DATA PATTERN
13839                                   ;#TO VERIFY THAT READING AND WRITING OF OTHER
13840                                   ;#LOCATIONS DID NOT CHANGE DATA.
13841  033260                   $XZ
13842                                   ;;*****************************************************************
13843
13844  033260                   $TSTN
13845                                   ;  TEST 63
13846                                   ;--------------
13847  033260  012737  000063  001202   TST63:  MOV     #63,$TSTNM              ; LOAD THE NO. OF THIS TEST
13848  033266  012737  034064  001442           MOV     #TST64,NEXT             ; POINT TO THE START OF NEXT TEST.
13849                                                            ;R1 CONTAINS BASE KMC11 ADDRESS
13850  033274  004737  035536                   JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
13851  033300  033314                   MCT63
13852  033302  104022                   ERROR   22                              ;TIME OUT ERROR...
13853  033304  012706  001200                   MOV     #STACK,SP               ;RESET STACK...
13854  033310  000177  146126                   JMP     @NEXT                   ;GO TO NEXT TEST...
13855  033314                   MCT63:
13856  033314                   18$:    MOVE    # 0,BREG                       ;START WITH ZERO.
13857          000000                   MICPC=MICPC+1
13858  033314  000400                   .WORD   .$.
13859  033316                           MOVE    BREG,SPAD <0>   ;
13860          000001                   MICPC=MICPC+1
13861  033316  063220                   .WORD   .$.
13862  033320                           MOVE    BREG,SPAD <1>   ;
13863          000002                   MICPC=MICPC+1
13864  033320  063221                   .WORD   .$.
13865  033322                           MOVE    BREG,SPAD <16>  ;FOR RETURN ADDRESS PURPOSES...
13866          000003                   MICPC=MICPC+1
13867  033322  063236                   .WORD   .$.
13868
13869  033324                   1S:     MOVE    # 0,MLR ;MAR←0.
13870          000004                   MICPC=MICPC+1
13871  033324  010000                   .WORD   .$.
13872  033326                           MOVE    # 0,MPR ;
13873          000005                   MICPC=MICPC+1
13874  033326  004000                   .WORD   .$.
13875  033330                           MOVE    BREG,OUT1 <CSR0>        ;WRITE IBUS* REGISTERS.
13876          000006                   MICPC=MICPC+1
13877  033330  061220                   .WORD   .$.
13878  033332                           MOVE    BREG,OUT1 <CSR2>        ;WRITE IBUS* REGISTERS.
13879          000007                   MICPC=MICPC+1
13880  033332  061222                   .WORD   .$.
13881  033334                           MOVE    BREG,OUT1 <CSR3>        ;WRITE IBUS* REGISTERS.
13882          000010                   MICPC=MICPC+1
13883  033334  061223                   .WORD   .$.
13884  033336                           MOVE    BREG,OUT1 <CSR4>        ;WRITE IBUS* REGISTERS.
13885          000011                   MICPC=MICPC+1
13886  033336  061224                   .WORD   .$.
13887  033340                           MOVE    BREG,OUT1 <CSR5>        ;WRITE IBUS* REGISTERS.
13888          000012                   MICPC=MICPC+1
13889  033340  061225                   .WORD   .$.
13890
13891  033342                           MOVE    BREG,OUT1 <CSR6>        ;WRITE IBUS* REGISTERS.
```

```
13892                   000013                  MICPC=MICPC+1
13893 033342            061226                  .WORD   .$.
13894 033344                                    MOVE    BREG,OUT1 (CSR7)        ;WRITE IBUS* REGISTERS.
13895                   000014                  MICPC=MICPC+1
13896 033344            061227                  .WORD   .$.
13897                                   ;:*
13898                                   ;:*
13899 033346                                    MOVE    BREG,OUT0 (0)   ;WRITE IBUS REGISTERS.
13900                   000015                  MICPC=MICPC+1
13901 033346            062220                  .WORD   .$.
13902 033350                                    MOVE    BREG,OUT0 (1)   ;WRITE IBUS REGISTERS.
13903                   000016                  MICPC=MICPC+1
13904 033350            062221                  .WORD   .$.
13905 033352                                    MOVE    BREG,OUT0 (2)   ;WRITE IBUS REGISTERS.
13906                   000017                  MICPC=MICPC+1
13907 033352            062222                  .WORD   .$.
13908 033354                                    MOVE    BREG,OUT0 (3)   ;WRITE IBUS REGISTERS.
13909                   000020                  MICPC=MICPC+1
13910 033354            062223                  .WORD   .$.
13911 033356                                    MOVE    BREG,OUT0 (4)   ;WRITE IBUS REGISTERS.
13912                   000021                  MICPC=MICPC+1
13913 033356            062224                  .WORD   .$.
13914 033360                                    MOVE    BREG,OUT0 (5)   ;WRITE IBUS REGISTERS.
13915                   000022                  MICPC=MICPC+1
13916 033360            062225                  .WORD   .$.
13917 033362                                    MOVE    BREG,OUT0 (6)   ;WRITE IBUS REGISTERS.
13918                   000023                  MICPC=MICPC+1
13919 033362            062226                  .WORD   .$.
13920 033364                                    MOVE    BREG,OUT0 (7)   ;WRITE IBUS REGISTERS.
13921                   000024                  MICPC=MICPC+1
13922 033364            062227                  .WORD   .$.
13923                                   ;:*
13924                                   ;:*
13925                                   ;:*
13926 033366                                    MOVE    BREG,SPAD (3)   ;WRITE SPAD LOCATIONS.
13927                   000025          ;WRITE SPAD LOCATIONS.
13928 033366            063223                  MICPC=MICPC+1
13929 033370                                    .WORD   .$.
13930                   000026                  MOVE    BREG,SPAD (4)   ;WRITE SPAD LOCATIONS.
13931 033370            063224                  MICPC=MICPC+1
13932 033372                                    .WORD   .$.
13933                   000027                  MOVE    BREG,SPAD (5)   ;WRITE SPAD LOCATIONS.
13934 033372            063225                  MICPC=MICPC+1
13935 033374                                    .WORD   .$.
13936                   000030                  MOVE    BREG,SPAD (6)   ;WRITE SPAD LOCATIONS.
13937 033374            063226                  MICPC=MICPC+1
13938 033376                                    .WORD   .$.
13939                   000031                  MOVE    BREG,SPAD (7)   ;WRITE SPAD LOCATIONS.
13940 033376            063227                  MICPC=MICPC+1
13941 033400                                    .WORD   .$.
13942                   000032                  MOVE    BREG,SPAD (10)  ;WRITE SPAD LOCATIONS.
13943 033400            063230                  MICPC=MICPC+1
13944 033402                                    .WORD   .$.
13945                   000033                  MOVE    BREG,SPAD (11)  ;WRITE SPAD LOCATIONS.
13946 033402            063231                  MICPC=MICPC+1
13947 033404                                    .WORD   .$.
                                                MOVE    BREG,SPAD (12)  ;WRITE SPAD LOCATIONS.
```

```
13948                000034                          MICPC=MICPC+1
13949  033404  063232                                .WORD   .$.
13950  033406                                        MOVE    BREG,SPAD <13>  ;WRITE SPAD LOCATIONS.
13951                000035                          MICPC=MICPC+1
13952  033406  063233                                .WORD   .$.
13953  033410                                        MOVE    BREG,SPAD <14>  ;WRITE SPAD LOCATIONS.
13954                000036                          MICPC=MICPC+1
13955  033410  063234                                .WORD   .$.
13956  033412                                        MOVE    BREG,SPAD <15>  ;WRITE SPAD LOCATIONS.
13957                000037                          MICPC=MICPC+1
13958  033412  063235                                .WORD   .$.
13959                                        ;:*
13960                                        ;:*
13961  033414                                2$:     MOVE    BREG,MEM MARINC ;WRITE MEMORY LOCATION.
13962                000040                          MICPC=MICPC+1
13963  033414  076620                                .WORD   .$.
13964  033416                                        $INC    SPAD <0>                ;UPDATE COUNT.
13965                000041                          MICPC=MICPC+1
13966  033416  063060                        .WORD   .$.!.DSP
13967  033420                                        $ADC    SPAD <1>                ;UPDATE COUNT.
13968                000042                          MICPC=MICPC+1
13969  033420  063101                        .WORD   .$.!.DSP
13970  033422                                        MOVE    BREG,SPAD <2>   ;IS IT DONE?
13971                000043                          MICPC=MICPC+1
13972  033422  063222                                .WORD   .$.
13973  033424                                        MOVE    # 4,BREG                ;IS IT DONE?
13974                000044                          MICPC=MICPC+1
13975  033424  000404                                .WORD   .$.
13976  033426                                        $IFHIS  BREG,SPAD <1> 3$        ;NO, CONTINUE.
13977
13978
13979  033426                                        SUB2C   SPAD <1>,BREG,NOP
13980                000045                          MICPC=MICPC+1
13981  033426  060361                                .WORD   .$.
13982  033430                                        BC      64$
13983                000046                          MICPC=MICPC+1
13984  033430  101050                                .WORD   .$.
13985  033432                                        $BR     3$
13986                000047                          MICPC=MICPC+1
13987  033432  100452                                .WORD   .$.
13988  033434                                64$:
13989  033434                                        MOVE    SPAD <2>,BREG
13990                000050                          MICPC=MICPC+1
13991  033434  060602                                .WORD   .$.
13992  033436                                        $BR     4$              ;YES, WRITE THE NEXT.
13993                000051                          MICPC=MICPC+1
13994  033436  100454                                .WORD   .$.
13995  033440                                3$:     MOVE    SPAD <2>,BREG   ;RESTORE BREG.
13996                000052                          MICPC=MICPC+1
13997  033440  060602                                .WORD   .$.
13998  033442                                        $BR     2$              ;WRITE NEXT MEMORY LOCATION
13999                000053                          MICPC=MICPC+1
14000  033442  100440                                .WORD   .$.
14001  033444                                4$:
14002  033444                                        $DEC    SPAD <2>                ;WAS IT ZERO?
14003                000054                          MICPC=MICPC+1
```

```
14004  033444  063162                .WORD   .$.!.DSP
14005  033446                                BZ      5$                      ;YES, NOW LOAD 377.
14006          000055                        MICPC=MICPC+1
14007  033446  101462                .WORD   .$.
14008  033450                                MOVE    BREG,SPAD <2>           ;WAS IT 377?
14009          000056                        MICPC=MICPC+1
14010  033450  063222                .WORD   .$.
14011  033452                                $INC    SPAD <2>                        ;WAS IT 377?
14012          000057                        MICPC=MICPC+1
14013  033452  063062                .WORD   .$.!.DSP
14014  033454                                BC      6$                      ;YES, NOW LOAD 252.
14015          000060                        MICPC=MICPC+1
14016  033454  101064                .WORD   .$.
14017  033456                                SBR     9$                      ;DONE START READING.
14018          000061                        MICPC=MICPC+1
14019  033456  100471                .WORD   .$.
14020  033460                        5$:     MOVE    # 377,BREG              ;LOAD 377.
14021          000062                        MICPC=MICPC+1
14022  033460  000777                .WORD   .$.
14023  033462                                SBR     7$                      ;
14024          000063                        MICPC=MICPC+1
14025  033462  100465                .WORD   .$.
14026  033464                        6$:     MOVE    # 252,BREG              ;LOAD 252
14027          000064                        MICPC=MICPC+1
14028  033464  000652                .WORD   .$.
14029  033466                        7$:
14030  033466                                MOVE    # 0,SPAD <0>             ;RESET SPAD 0.
14031          000065                        MICPC=MICPC+1
14032  033466  003000                .WORD   .$.
14033  033470                                MOVE    # 1,SPAD <1>             ;RESET SPAD 1.
14034          000066                        MICPC=MICPC+1
14035  033470  003001                .WORD   .$.
14036  033472                                $DEC    SPAD <1>                        ;RESET SPAD 1.
14037          000067                        MICPC=MICPC+1
14038  033472  063161                .WORD   .$.!.DSP
14039  033474                                SBR     1$                      ;CONTINUE
14040          000070                        MICPC=MICPC+1
14041  033474  100404                .WORD   .$.
14042                                ;:*
14043                                ;:*     NOW GO BACK AND READ THEM.
14044                                ;:*
14045  033476                        9$:     MOVE    BREG,SPAD <2>           ;SET UP FOR CHECKING.
14046          000071                        MICPC=MICPC+1
14047  033476  063222                .WORD   .$.
14048  033500                                MOVE    INP1 <CSR0>,SPAD <1>    ;GET IBUS* REGISTER
14049          000072                        MICPC=MICPC+1
14050  033500  123001                .WORD   .$.
14051  033502                                CALL    CMPRE                   ;VERIFY IT.
14052  033502                                MOVE    # <MICPC+3>,BREG
14053          000073                        MICPC=MICPC+1
14054  033502  000475                .WORD   .$.
14055  033504                                SBR     CMPRE
14056          000074                        MICPC=MICPC+1
14057  033504  100650                .WORD   .$.
14058  033506                                MOVE    INP1 <CSR2>,SPAD <1>    ;READ NEXT IBUS* REG.
14059          000075                        MICPC=MICPC+1
```

```
14060  033506  123041                  .WORD   .$.
14061  033510                          CALL    CMPRE           ;VERIFY IT.
14062  033510                          MOVE    # <MICPC+3>,BREG
14063          000076                  MICPC=MICPC+1
14064  033510  000500                  .WORD   .$.
14065  033512                          $BR     CMPRE
14066          000077                  MICPC=MICPC+1
14067  033512  100650                  .WORD   .$.
14068  033514                          MOVE    INP1 <CSR3>,SPAD <1>    ;READ NEXT IBUS* REG.
14069          000100                  MICPC=MICPC+1
14070  033514  123061                  .WORD   .$.
14071  033516                          CALL    CMPRE           ;VERIFY IT.
14072  033516                          MOVE    # <MICPC+3>,BREG
14073          000101                  MICPC=MICPC+1
14074  033516  000503                  .WORD   .$.
14075  033520                          $BR     CMPRE
14076          000102                  MICPC=MICPC+1
14077  033520  100650                  .WORD   .$.
14078  033522                          MOVE    INP1 <CSR4>,SPAD <1>    ;READ NEXT IBUS* REG.
14079          000103                  MICPC=MICPC+1
14080  033522  123101                  .WORD   .$.
14081  033524                          CALL    CMPRE           ;VERIFY IT.
14082  033524                          MOVE    # <MICPC+3>,BREG
14083          000104                  MICPC=MICPC+1
14084  033524  000506                  .WORD   .$.
14085  033526                          $BR     CMPRE
14086          000105                  MICPC=MICPC+1
14087  033526  100650                  .WORD   .$.
14088  033530                          MOVE    INP1 <CSR5>,SPAD <1>    ;READ NEXT IBUS* REG.
14089          000106                  MICPC=MICPC+1
14090  033530  123121                  .WORD   .$.
14091  033532                          CALL    CMPRE           ;VERIFY IT.
14092  033532                          MOVE    # <MICPC+3>,BREG
14093          000107                  MICPC=MICPC+1
14094  033532  000511                  .WORD   .$.
14095  033534                          $BR     CMPRE
14096          000110                  MICPC=MICPC+1
14097  033534  100650                  .WORD   .$.
14098  033536                          MOVE    INP1 <CSR6>,SPAD <1>    ;READ NEXT IBUS* REG.
14099          000111                  MICPC=MICPC+1
14100  033536  123141                  .WORD   .$.
14101  033540                          CALL    CMPRE           ;VERIFY IT.
14102  033540                          MOVE    # <MICPC+3>,BREG
14103          000112                  MICPC=MICPC+1
14104  033540  000514                  .WORD   .$.
14105  033542                          $BR     CMPRE
14106          000113                  MICPC=MICPC+1
14107  033542  100650                  .WORD   .$.
14108  033544                          MOVE    INP1 <CSR7>,SPAD <1>    ;READ NEXT IBUS* REG.
14109          000114                  MICPC=MICPC+1
14110  033544  123161                  .WORD   .$.
14111  033546                          CALL    CMPRE           ;VERIFY IT.
14112  033546                          MOVE    # <MICPC+3>,BREG
14113          000115                  MICPC=MICPC+1
14114  033546  000517                  .WORD   .$.
14115  033550                          $BR     CMPRE
```

```
14116                  000116                  MICPC=MICPC+1
14117  033550          100650                  .WORD   .$.
14118                                  ;:*
14119                                  ;:*
14120  033552                          MOVE    INPO <0>,SPAD <1>       ;GET IBUS REG.
14121                  000117          MICPC=MICPC+1
14122  033552          023001          .WORD   $.
14123  033554                          CALL    CMPRE                   ;VERIFY IT.
14124  033554                          MOVE    @ <MICPC+3>,BREG
14125                  000120          MICPC=MICPC+1
14126  033554          000522          .WORD   .$.
14127  033556                          SBR     CMPRE
14128                  000121          MICPC=MICPC+1
14129  033556          100650          .WORD   $.
14130  033560                          MOVE    INPO <1>,SPAD <1>       ;GET NEXT IBUS-REGISTER.
14131                  000122          MICPC=MICPC+1
14132  033560          023021          .WORD   .$.
14133  033562                          CALL    CMPRE                   ;VERIFY IT.
14134  033562                          MOVE    @ <MICPC+3>,BREG
14135                  000123          MICPC=MICPC+1
14136  033562          000525          .WORD   .$.
14137  033564                          SBR     CMPRE
14138                  000124          MICPC=MICPC+1
14139  033564          100650          .WORD   $.
14140  033566                          MOVE    INPO <2>,SPAD <1>       ;GET NEXT IBUS+REG.
14141                  000125          MICPC=MICPC+1
14142  033566          023041          .WORD   .$.
14143  033570                          CALL    CMPRE                   ;VERIFY IT.
14144  033570                          MOVE    @ <MICPC+3>,BREG
14145                  000126          MICPC=MICPC+1
14146  033570          000530          .WORD   .$.
14147  033572                          SBR     CMPRE
14148                  000127          MICPC=MICPC+1
14149  033572          100650          .WORD   $.
14150  033574                          MOVE    INPO <3>,SPAD <1>       ;GET NEXT IBUS+REG.
14151                  000130          MICPC=MICPC+1
14152  033574          023061          .WORD   .$.
14153  033576                          CALL    CMPRE                   ;VERIFY IT.
14154  033576                          MOVE    @ <MICPC+3>,BREG
14155                  000131          MICPC=MICPC+1
14156  033576          000533          .WORD   .$.
14157  033600                          SBR     CMPRE
14158                  000132          MICPC=MICPC+1
14159  033600          100650          .WORD   $.
14160  033602                          MOVE    INPO <4>,SPAD <1>       ;GET NEXT IBUS+REG.
14161                  000133          MICPC=MICPC+1
14162  033602          023101          .WORD   .$.
14163  033604                          CALL    CMPRE                   ;VERIFY IT.
14164  033604                          MOVE    @ <MICPC+3>,BREG
14165                  000134          MICPC=MICPC+1
14166  033604          000536          .WORD   .$.
14167  033606                          SBR     CMPRE
14168                  000135          MICPC=MICPC+1
14169  033606          100650          .WORD   $.
14170  033610                          MOVE    INPO <5>,SPAD <1>       ;GET NEXT IBUS+REG.
14171                  000136          MICPC=MICPC+1
```

```
14172  033610  023121                          .WORD   .$.
14173  033612                                  CALL    CMPRE           ;VERIFY IT.
14174  033612                                  MOVE    # <MICPC+3>,BREG
14175          000137                          MICPC=MICPC+1
14176  033612  000541                          .WORD   .$.
14177  033614                                  SBR     CMPRE
14178          000140                          MICPC=MICPC+1
14179  033614  100650                          .WORD   .$.
14180  033616                                  MOVE    INPO <6>,SPAD <1>       ;GET NEXT IBUS+REG.
14181          000141                          MICPC=MICPC+1
14182  033616  023141                          .WORD   .$.
14183  033620                                  CALL    CMPRE           ;VERIFY IT
14184  033620                                  MOVE    # <MICPC+3>,BREG
14185          000142                          MICPC=MICPC+1
14186  033620  000544                          .WORD   .$.
14187  033622                                  SBR     CMPRE
14188          000143                          MICPC=MICPC+1
14189  033622  100650                          .WORD   .$.
14190  033624                                  MOVE    INPO <7>,SPAD <1>       ;GET NEXT IBUS+REG.
14191          000144                          MICPC=MICPC+1
14192  033624  023161                          .WORD   .$.
14193  033626                                  CALL    CMPRE           ;VERIFY IT.
14194  033626                                  MOVE    # <MICPC+3>,BREG
14195          000145                          MICPC=MICPC+1
14196  033626  000547                          .WORD   .$.
14197  033630                                  SBR     CMPRE
14198          000146                          MICPC=MICPC+1
14199  033630  100650                          .WORD   .$.
14200
14201                                  ;:#
14202                                  ;:#
14203  033632                          ;:#     MOVE    SPAD <3>,BREG   ;GET SPAD LOCATION.
14204          000147                          MICPC=MICPC+1
14205  033632  060603                          .WORD   .$.
14206  033634                                  MOVE    BREG,SPAD <1>   ;
14207          000150                          MICPC=MICPC+1
14208  033634  063221                          .WORD   .$.
14209  033636                                  CALL    CMPRE           ;VERIFY IT.
14210  033636                                  MOVE    # <MICPC+3>,BREG
14211          000151                          MICPC=MICPC+1
14212  033636  000553                          .WORD   .$.
14213  033640                                  SBR     CMPRE
14214          000152                          MICPC=MICPC+1
14215  033640  100650                          .WORD   .$.
14216  033642                                  MOVE    SPAD <4>,BREG   ;GET NEXT SPAD LOCATION.
14217          000153                          MICPC=MICPC+1
14218  033642  060604                          .WORD   .$.
14219  033644                                  MOVE    BREG,SPAD <1>   ;
14220          000154                          MICPC=MICPC+1
14221  033644  063221                          .WORD   .$.
14222  033646                                  MOVE    SPAD <5>,BREG   ;GET NEXT SPAD LOCATION.
14223          000155                          MICPC=MICPC+1
14224  033646  060605                          .WORD   .$.
14225  033650                                  MOVE    BREG,SPAD <1>   ;
14226          000156                          MICPC=MICPC+1
14227  033650  063221                          .WORD   .$.
```

```
14228  033652                  CALL    CMPRE
14229  033652                  MOVE    # <MICPC+3>,BREG    ;VERIFY IT.
14230          000157          MICPC=MICPC+1
14231  033652  000561          .WORD   .$.
14232  033654                  SBR     CMPRE
14233          000160          MICPC=MICPC+1
14234  033654  100650          .WORD   .$.
14235  033656                  MOVE    SPAD <6>,BREG   ;GET NEXT SPAD LOCATION.
14236          000161          MICPC=MICPC+1
14237  033656  060606          .WORD   .$.
14238  033660                  MOVE    BREG,SPAD <1>   ;
14239          000162          MICPC=MICPC+1
14240  033660  063221          .WORD   .$.
14241  033662                  CALL    CMPRE
14242  033662                  MOVE    # <MICPC+3>,BREG    ;VERIFY IT.
14243          000163          MICPC=MICPC+1
14244  033662  000565          .WORD   .$.
14245  033664                  SBR     CMPRE
14246          000164          MICPC=MICPC+1
14247  033664  100650          .WORD   .$.
14248  033666                  MOVE    SPAD <7>,BREG   ;GET NEXT SPAD LOCATION.
14249          000165          MICPC=MICPC+1
14250  033666  060607          .WORD   .$.
14251  033670                  MOVE    BREG,SPAD <1>   ;
14252          000166          MICPC=MICPC+1
14253  033670  063221          .WORD   .$.
14254  033672                  CALL    CMPRE
14255  033672                  MOVE    # <MICPC+3>,BREG    ;VERIFY IT.
14256          000167          MICPC=MICPC+1
14257  033672  000571          .WORD   .$.
14258  033674                  SBR     CMPRE
14259          000170          MICPC=MICPC+1
14260  033674  100650          .WORD   .$.
14261  033676                  MOVE    SPAD <10>,BREG  ;GET NEXT SPAD LOCATION.
14262          000171          MICPC=MICPC+1
14263  033676  060610          .WORD   .$.
14264  033700                  MOVE    BREG,SPAD <2>   ;
14265          000172          MICPC=MICPC+1
14266  033700  063222          .WORD   .$.
14267  033702                  CALL    CMPRE
14268  033702                  MOVE    # <MICPC+3>,BREG    ;VERIFY IT.
14269          000173          MICPC=MICPC+1
14270  033702  000575          .WORD   .$.
14271  033704                  SBR     CMPRE
14272          000174          MICPC=MICPC+1
14273  033704  100650          .WORD   .$.
14274  033706                  MOVE    SPAD <11>,BREG  ;GET NEXT SPAD LOCATION.
14275          000175          MICPC=MICPC+1
14276  033706  060611          .WORD   .$.
14277  033710                  MOVE    BREG,SPAD <1>   ;
14278          000176          MICPC=MICPC+1
14279  033710  063221          .WORD   .$.
14280  033712                  CALL    CMPRE
14281  033712                  MOVE    # <MICPC+3>,BREG    ;VERIFY IT.
14282          000177          MICPC=MICPC+1
14283  033712  000601          .WORD   .$.
```

```
14284  033714                    SBR     CMPRE
14285          000200            MICPC=MICPC+1
14286  033714  100650            .WORD   .$.
14287  033716                    MOVE    SPAD (12),BREG  ;GET NEXT SPAD LOCATION.
14288          000201            MICPC=MICPC+1
14289  033716  060612            .WORD   .$.
14290  033720                    MOVE    BREG,SPAD (1)   ;
14291          000202            MICPC=MICPC+1
14292  033720  063221            .WORD   .$.
14293  033722                    CALL    CMPRE           ;VERIFY IT.
14294  033722                    MOVE    # <MICPC+3),BREG
14295          000203            MICPC=MICPC+1
14296  033722  000605            .WORD   .$.
14297  033724                    SBR     CMPRE
14298          000204            MICPC=MICPC+1
14299  033724  100650            .WORD   .$.
14300  033726                    MOVE    SPAD (13),BREG  ;GET NEXT SPAD LOCATION.
14301          000205            MICPC=MICPC+1
14302  033726  060613            .WORD   .$.
14303  033730                    MOVE    BREG,SPAD (1)   ;
14304          000206            MICPC=MICPC+1
14305  033730  063221            .WORD   .$.
14306  033732                    CALL    CMPRE           ;VERIFY IT.
14307  033732                    MOVE    # <MICPC+3),BREG
14308          000207            MICPC=MICPC+1
14309  033732  000611            .WORD   .$.
14310  033734                    SBR     CMPRE
14311          000210            MICPC=MICPC+1
14312  033734  100650            .WORD   .$.
14313  033736                    MOVE    SPAD (14),BREG  ;GET NEXT SPAD LOCATION.
14314          000211            MICPC=MICPC+1
14315  033736  060614            .WORD   .$.
14316  033740                    MOVE    BREG,SPAD (1)   ;
14317          000212            MICPC=MICPC+1
14318  033740  063221            .WORD   .$.
14319  033742                    CALL    CMPRE           ;VERIFY IT.
14320  033742                    MOVE    # <MICPC+3),BREG
14321          000213            MICPC=MICPC+1
14322  033742  000615            .WORD   .$.
14323  033744                    SBR     CMPRE
14324          000214            MICPC=MICPC+1
14325  033744  100650            .WORD   .$.
14326  033746                    MOVE    SPAD (15),BREG  ;GET NEXT SPAD LOCATION.
14327          000215            MICPC=MICPC+1
14328  033746  060615            .WORD   .$.
14329  033750                    MOVE    BREG,SPAD (1)   ;
14330          000216            MICPC=MICPC+1
14331  033750  063221            .WORD   .$.
14332  033752                    CALL    CMPRE           ;VERIFY IT.
14333  033752                    MOVE    # <MICPC+3),BREG
14334          000217            MICPC=MICPC+1
14335  033752  000621            .WORD   .$.
14336  033754                    SBR     CMPRE
14337          000220            MICPC=MICPC+1
14338  033754  100650            .WORD   .$.
14339
                        ;:*
```

```
14340                                   ;:*
14341                                   ;:*
14342 033756                                    MOVE    # 0,MLR          ;
14343        000221                             MICPC=MICPC+1
14344 033756 010000                             .WORD   .$.
14345 033760                                    MOVE    # 0,MPR          ;
14346        000222                             MICPC=MICPC+1
14347 033760 004000                             .WORD   .$.
14348 033762                                    MOVE    BREG,SPAD <0>    ;SET THE COUNTER.
14349        000223                             MICPC=MICPC+1
14350 033762 063220                             .WORD   .$.
14351 033764                                    MOVE    # 3,SPAD <3>     ;SET THE COUNTER.
14352        000224                             MICPC=MICPC+1
14353 033764 003003                             .WORD   .$.
14354 033766                            125:    MOVE    MEM,BREG MARINC ;GET THE CONTENTS OF MEMORY LOCATION.
14355        000225                             MICPC=MICPC+1
14356 033766 054620                             .WORD   .$.
14357 033770                                    $IFEQ   BREG,SPAD <2> 15$       ;BRANCH IF GOOD.
14358
14359
14360 033770                                    SUB2C   SPAD <2>,BREG,NOP
14361        000226                             MICPC=MICPC+1
14362 033770 060362                             .WORD   .$.
14363 033772                                    BZ      15$
14364        000227                             MICPC=MICPC+1
14365 033772 101637                             .WORD   .$.
14366 033774                                    MOVE    MEM,OUT1 <CSR4> ;GOOD DATA
14367        000230                             MICPC=MICPC+1
14368 033774 041224                             .WORD   .$.
14369 033776                                    MOVE    BREG,OUT1 <CSR5>        ;BAD DATA
14370        000231                             MICPC=MICPC+1
14371 033776 061225                             .WORD   .$.
14372 034000                                    MOVE    # 20,BREG       ;
14373        000232                             MICPC=MICPC+1
14374 034000 000420                             .WORD   .$.
14375 034002                                    MOVE    BREG,OUT1 <CSR3>        ;TYPE OF ERROR.
14376        000233                             MICPC=MICPC+1
14377 034002 061223                             .WORD   .$.
14378 034004                                    CALL    ERROR1
14379 034004                                    MOVE    # <MICPC+3>,BREG;DATA ERROR.
14380        000234                             MICPC=MICPC+1
14381 034004 000636                             .WORD   .$.
14382 034006                                    SBR     ERROR1
14383        000235                             MICPC=MICPC+1
14384 034006 104401                             .WORD   .$.
14385 034010                                    SBR     18$             ;LOOP ON ERROR.
14386        000236                             MICPC=MICPC+1
14387 034010 100400                             .WORD   .$.
14388 034012                            15$:    $INC    SPAD <0>                ;COUNT BY ONE.
14389        000237                             MICPC=MICPC+1
14390 034012 063060                     .WORD   .$.!.DSP
14391 034014                                    BC      16$             ;COUNT BY ONE.
14392        000240                             MICPC=MICPC+1
14393 034014 101242                             .WORD   .$.
14394 034016                                    SBR     12$             ;DO THE NEXT.
14395        000241                             MICPC=MICPC+1
```

```
14396 034016 100625                    .WORD   .$.
14397 034020                   16$:    SDEC    SPAD <3>                ;IS IT DONE??
14398        000242                    MICPC=MICPC+1
14399 034020 063163            .WORD   .$.!.DSP
14400 034022                           BZ      17$                     ;YES,SCOPE IT.
14401        000243                    MICPC=MICPC+1
14402 034022 101645                    .WORD   .$.
14403 034024                           SBR     12$                     ;ELSE, DO NEXT.
14404        000244                    MICPC=MICPC+1
14405 034024 100625                    .WORD   .$.
14406 034026                   17$:    CALL    SCPE                    ;SCOPE THE TEST...
14407 034026                           MOVE    # <MICPC+3>,BREG
14408        000245                    MICPC=MICPC+1
14409 034026 000647                    .WORD   .$.
14410 034030                           SBR     SCPE
14411        000246                    MICPC=MICPC+1
14412 034030 104454                    .WORD   .$.
14413 034032                           SBR     10$                     ;DO THE NEXT ITERATION.
14414        000247                    MICPC=MICPC+1
14415 034032 100400                    .WORD   .$.
14416 034034                   CMPRE:  MOVE    BREG,SPAD <17>          ;SAVE THE RETURN ADDRESS.
14417        000250                    MICPC=MICPC+1
14418 034034 063237                    .WORD   .$.
14419 034036                           MOVE    SPAD <1>,BREG           ;GET FOUND IN BREG.
14420        000251                    MICPC=MICPC+1
14421 034036 060601                    .WORD   .$.
14422 034040                           SIFEQ   BREG,SPAD <2> 33$       ;BRANCH IF GOOD.
14423
14424
14425 034040                           SUB2C   SPAD <2>,BREG,NOP
14426        000252                    MICPC=MICPC+1
14427 034040 060362                    .WORD   .$.
14428 034042                           BZ      33$
14429        000253                    MICPC=MICPC+1
14430 034042 101663                    .WORD   .$.
14431 034044                           MOVE    BREG,OUT1 <CSR5>        ;BAD DATA.
14432        000254                    MICPC=MICPC+1
14433 034044 061225                    .WORD   .$.
14434 034046                           MOVE    SPAD <2>,BREG   ;
14435        000255                    MICPC=MICPC+1
14436 034046 060602                    .WORD   .$.
14437 034050                           MOVE    BREG,OUT1 <CSR4>        ;GOOD DATA.
14438        000256                    MICPC=MICPC+1
14439 034050 061224                    .WORD   .$.
14440 034052                           MOVE    # 20,BREG       ;
14441        000257                    MICPC=MICPC+1
14442 034052 000420                    .WORD   .$.
14443 034054                           MOVE    BREG,OUT1 <CSR3>        ;TYPE OF ERROR.
14444        000260                    MICPC=MICPC+1
14445 034054 061223                    .WORD   .$.
14446 034056                           CALL    ERROR1                  ;REPORT ERROR
14447 034056                           MOVE    # <MICPC+3>,BREG
14448        000261                    MICPC=MICPC+1
14449 034056 000663                    .WORD   .$.
14450 034060                           SBR     ERROR1
14451        000262                    MICPC=MICPC+1
```

```
14452  034060  104401                      .WORD   .$.
14453  034062                      33$:
14454  034062                              SBR     SPAD (17) PAGE0
14455          000263                       MICPC=MICPC+1
14456  034062  160617                      .WORD   .$.
14457  034064                      SHLRSR
14458  034064                      SXZ
14459
14460
14461                                       ;*************************** TEST 64 ****************************
14462                                       ;* HELL RAISER TEST...
14463                                       ;* ONLY TO TEST MPRAM CONTROL LOGIC...
14464                                       ;* NOT FOR MAINTENENCE PURPOSE...
14465  034064                      SXZ
14466                                       ;;*************************************************************************
14467
14468  034064                      STSTN
14469                                       ;  TEST 64
14470                                       ;-------------
14471  034064  012737  000064  001202  TST64:  MOV     #64,$TSTNM              ; LOAD THE NO. OF THIS TEST
14472  034072  012737  034314  001442        MOV     #TST65,NEXT             ; POINT TO THE START OF NEXT TEST.
14473                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
14474  034100  004737  035536              JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
14475  034104  034120                      MCT64
14476  034106  104022                      ERROR   22                      ;TIME OUT ERROR...
14477  034110  012706  001200              MOV     #STACK,SP               ;RESET STACK...
14478  034114  000177  145322              JMP     @NEXT                   ;GO TO NEXT TEST...
14479  034120                      MCT64:
14480  034120                      1$:     MOVE    # 0,BREG         ;
14481          000000                      MICPC=MICPC+1
14482  034120  000400                      .WORD   .$.
14483  034122                              MOVE    # 0,MLR          ;CLEAR MAR.
14484          000001                      MICPC=MICPC+1
14485  034122  010000                      .WORD   .$.
14486  034124                              MOVE    # 0,MPR          ;CLEAR MAR.
14487          000002                      MICPC=MICPC+1
14488  034124  004000                      .WORD   .$.
14489  034126                              MOVE    BREG,OUT1 (CSR4)        ;CLEAR BSEL4.
14490          000003                      MICPC=MICPC+1
14491  034126  061224                      .WORD   .$.
14492  034130                              MOVE    BREG,OUT1 (CSR5)        ;CLEAR BSEL5.
14493          000004                      MICPC=MICPC+1
14494  034130  061225                      .WORD   .$.
14495  034132                              MOVE    BREG,OUT0 (0)           ;CLEAR IN DATA LB.
14496          000005                      MICPC=MICPC+1
14497  034132  062220                      .WORD   .$.
14498  034134                              MOVE    BREG,OUT0 (1)           ;CLEAR IN DATA HB.
14499          000006                      MICPC=MICPC+1
14500  034134  062221                      .WORD   .$.
14501  034136                              MOVE    BREG,OUT0 (4)           ;CLEAR IN BA LB.
14502          000007                      MICPC=MICPC+1
14503  034136  062224                      .WORD   .$.
14504  034140                              MOVE    BREG,OUT0 (5)           ;CLEAR IN BA HB.
14505          000010                      MICPC=MICPC+1
14506  034140  062225                      .WORD   .$.
14507  034142                              MOVE    BREG,SPAD (16)  ;RETURN ADDRESS PURPOSE.
```

```
14508                000011                          MICPC=MICPC+1
14509  034142 063236                                 .WORD   .$.
14510  034144                                        MOVE    # <HELDAT&377>,BREG     ;SET IN+BA LB.
14511                000012                          MICPC=MICPC+1
14512  034144 000712                                 .WORD   .$.
14513  034146                                        MOVE    BREG,OUT0 <4>           ;
14514                000013                          MICPC=MICPC+1
14515  034146 062224                                 .WORD   .$.
14516  034150                                        MOVE    # <HELDAT/400&377>,BREG ;SET IN+BA+HB.
14517                000014                          MICPC=MICPC+1
14518  034150 000470                                 .WORD   .$.
14519  034152                                        MOVE    BREG,OUT0 <5>           ;
14520                000015                          MICPC=MICPC+1
14521  034152 062225                                 .WORD   .$.
14522  034154                                        MOVE    # 022,MEM       ;
14523                000016                          MICPC=MICPC+1
14524  034154 002422                                 .WORD   .$.
14525  034156                                        MOVE    MEM,OUT1 <CSR2> ;SET FOR HELL RAISER...
14526                000017                          MICPC=MICPC+1
14527  034156 041222                                 .WORD   .$.
14528  034160                                        MOVE    # 357,BREG      ;GET THE MASK.
14529                000020                          MICPC=MICPC+1
14530  034160 000757                                 .WORD   .$.
14531  034162                                        MOVE    INP1 <CSR11>,SPAD <0>   ;GET UPMS REGISTER.
14532                000021                          MICPC=MICPC+1
14533  034162 123220                                 .WORD   .$.
14534  034164                                        AND     BREG,SPAD <0>,SPAD <0>  ;
14535                000022                          MICPC=MICPC+1
14536  034164 063260                                 .WORD   .$.
14537  034166                                        MOVE    # 300,BREG              ;SET BR+RQ & VCTR:=XX4.
14538                000023                          MICPC=MICPC+1
14539  034166 000700                                 .WORD   .$.
14540  034170                                        OR      BREG,SPAD <0>,OUT1 <11> ;
14541                000024                          MICPC=MICPC+1
14542  034170 061311                                 .WORD   .$.
14543  034172                        2$:             MOVE    INP1 <CSR11>,BREG       ;IS BR+RQ GRANTED???
14544                000025                          MICPC=MICPC+1
14545  034172 120620                                 .WORD   .$.
14546  034174                                        BR7     2$                      ;NO,SPIN ON IT.
14547                000026                          MICPC=MICPC+1
14548  034174 103425                                 .WORD   .$.
14549  034176                                        MOVE    # 0,BREG                ;
14550                000027                          MICPC=MICPC+1
14551  034176 000400                                 .WORD   .$.
14552  034200                                        MOVE    BREG,OUT1 <CSR11>       ;CLEARS VCTR:XX4.
14553                000030                          MICPC=MICPC+1
14554  034200 061231                                 .WORD   .$.
14555  034202                        3$:             MOVE    INP1 <CSR2>,BREG        ;IS PDP11 ALL SET???
14556                000031                          MICPC=MICPC+1
14557  034202 120440                                 .WORD   .$.
14558  034204                                        BR4     3$                      ;NO,WAIT FOR IT.
14559                000032                          MICPC=MICPC+1
14560  034204 103031                                 .WORD   .$.
14561  034206                        4$:             MOVE    # 001,BREG              ;SET NPR+RQ BIT.
14562                000033                          MICPC=MICPC+1
14563  034206 000401                                 .WORD   .$.
```

```
14564 034210                    MOVE    BREG,OUT1 <CSR10>       ;
14565         000034            MICPC=MICPC+1
14566 034210  061230            .WORD   .$.
14567 034212                5$: MOVE    INP1 <CSR10>,BREG       ;IS NPR DONE?
14568         000035            MICPC=MICPC+1
14569 034212  120600            .WORD   .$.
14570 034214                    BBO     6$
14571         000036            MICPC=MICPC+1
14572 034214  102044            .WORD   .$.
14573 034216                9$: MOVE    INP1 <CSR2>,BREG        ;IS CSR LOADED???
14574         000037            MICPC=MICPC+1
14575 034216  120440            .WORD   .$.
14576 034220                    BBI     4$                      ;NO WAIT FOR IT...
14577         000040            MICPC=MICPC+1
14578 034220  102433            .WORD   .$.
14579 034222                    MOVE    INP1 <CSR4>,SPAD <1>    ;
14580         000041            MICPC=MICPC+1
14581 034222  123101            .WORD   .$.
14582 034224                    MOVE    INP1 <CSR5>,SPAD <2>    ;
14583         000042            MICPC=MICPC+1
14584 034224  123122            .WORD   .$.
14585 034226                    SBR     12$                     ;GO CHECK DATA...
14586         000043            MICPC=MICPC+1
14587 034226  100451            .WORD   .$.
14588 034230                6$: MOVE    INP1 <CSR2>,BREG        ;IS CSR LOADED???
14589         000044            MICPC=MICPC+1
14590 034230  120440            .WORD   .$.
14591 034232                    BBI     5$                      ;NO,CHECK NPR DONE...
14592         000045            MICPC=MICPC+1
14593 034232  102435            .WORD   .$.
14594 034234                    MOVE    INP1 <CSR4>,SPAD <1>    ;TRANSFER TO SPAD 1.
14595         000046            MICPC=MICPC+1
14596 034234  123101            .WORD   .$.
14597 034236                    MOVE    INP1 <CSR5>,SPAD <2>    ;TRANSFER TO SPAD 2.
14598         000047            MICPC=MICPC+1
14599 034236  123122            .WORD   .$.
14600 034240                    SBR     5$                      ;GO CHECK NPR.
14601         000050            MICPC=MICPC+1
14602 034240  100435            .WORD   .$.
14603 034242                12$: MOVE   INP0 <0>,SPAD <0>       ;GET NPR DATA.
14604         000051            MICPC=MICPC+1
14605 034242  023000            .WORD   .$.
14606 034244                    MOVE    SPAD <1>,BREG           ;GET CSR DATA.
14607         000052            MICPC=MICPC+1
14608 034244  060601            .WORD   .$.
14609 034246                    $IFNE   BREG,SPAD <1>   15$     ;BR IF NO COMPARE.
14610
14611
14612 034246                    SUB2C   SPAD <1>,BREG,NOP
14613         000053            MICPC=MICPC+1
14614 034246  060361            .WORD   .$.
14615 034250                    BZ      64$
14616         000054            MICPC=MICPC+1
14617 034250  101456            .WORD   .$.
14618 034252                    $BR     15$
14619         000055            MICPC=MICPC+1
```

```
14620  034252  100462                  .WORD   .$.
14621  034254                  64$:
14622  034254                          MOVE    INPO (1),SPAD (0)       ;GET NPR DATA.
14623          000056                   MICPC=MICPC+1
14624  034254  023020                  .WORD   .$.
14625  034256                          MOVE    SPAD (2),BREG           ;GET CSR DATA.
14626          000057                   MICPC=MICPC+1
14627  034256  060602                  .WORD   .$.
14628  034260                          $IFEQ   BREG,SPAD (2)   16$     ;BRANCH IF GOOD.
14629
14630
14631  034260                          SUB2C   SPAD (2),BREG,NOP
14632          000060                   MICPC=MICPC+1
14633  034260  060362                  .WORD   .$.
14634  034262                          BZ      16$
14635          000061                   MICPC=MICPC+1
14636  034262  101472                  .WORD   .$.
14637  034264                  15$:    MOVE    BREG,OUT1 (CSR4)        ;CSR DATA.
14638          000062                   MICPC=MICPC+1
14639  034264  061224                  .WORD   .$.
14640  034266                          MOVE    SPAD (0),BREG           ;NPR DATA.
14641          000063                   MICPC=MICPC+1
14642  034266  060600                  .WORD   .$.
14643  034270                          MOVE    BREG,OUT1 (CSR4)        ;
14644          000064                   MICPC=MICPC+1
14645  034270  061224                  .WORD   .$.
14646  034272                          MOVE    # 25,BREG               ;ERROR TYPE.
14647          000065                   MICPC=MICPC+1
14648  034272  000425                  .WORD   .$.
14649  034274                          MOVE    BREG,OUT1 (CSR3)        ;
14650          000066                   MICPC=MICPC+1
14651  034274  061223                  .WORD   .$.
14652  034276                          CALL    ERROR1                  ;REPORT ERROR.
14653  034276                          MOVE    # (MICPC+3),BREG
14654          000067                   MICPC=MICPC+1
14655  034276  000471                  .WORD   .$.
14656  034300                          SBR     ERROR1
14657          000070                   MICPC=MICPC+1
14658  034300  104401                  .WORD   .$.
14659  034302                          SBR     1$                      ;LOOP ON ERROR.
14660          000071                   MICPC=MICPC+1
14661  034302  100400                  .WORD   .$.
14662  034304                  16$:    CALL    $CPE                    ;SCOPE THE TEST.
14663  034304                          MOVE    # (MICPC+3),BREG
14664          000072                   MICPC=MICPC+1
14665  034304  000474                  .WORD   .$.
14666  034306                          SBR     $CPE
14667          000073                   MICPC=MICPC+1
14668  034306  104454                  .WORD   .$.
14669  034310                          SBR     1$                      ;YES DO NEXT PASS.
14670          000074                   MICPC=MICPC+1
14671  034310  100400                  .WORD   .$.
14672  034312  125125          HELDAT: 125125
14673  034314                  $PWR1
14674  034314                  $XZ
14675
```

```
14676
14677                                   ;*************************** TEST 65 ****************************
14678                                   ;#FORCE POWER FAIL TEST.
14679                                   ;#SET FORCE POWER FAIL VERIFY THAT PROCESSOR TRAPS TO LOC 24.
14680                                   ;#GOING DOWN AND COMING UP. VERIFY ALSO THAT BUS INIT WAS
14681                                   ;#BLOCKED FROM GETTING TO KMC DURING THE POWER FAIL .
14682  034314                   SXZ
14683                                   ;;*************************************************************************
14684
14685
14686  034314                   STSTN
14687                                   ;  TEST 65
14688                                   ;-------------
14689  034314  012737  000065  001202  TST65:  MOV     #65,STSTNM              ; LOAD THE NO. OF THIS TEST
14690  034322  012737  003662  001442          MOV     #SEOP,NEXT              ; POINT TO THE END OF PASS HANDLER.
14691                                                                  ;R1 CONTAINS BASE KMC11 ADDRESS
14692  034330  004737  035536                  JSR     PC,LDVRWT               ;LOAD-VERIFY-WAIT.
14693  034334  034350                          MCT65
14694  034336  104022                          ERROR   22                      ;TIME OUT ERROR...
14695  034340  012706  001200                  MOV     #STACK,SP               ;RESET STACK...
14696  034344  000177  145072                  JMP     @NEXT                   ;GO TO NEXT TEST...
14697  034350                   MCT65:
14698  034350                   1$:    MOVE    # 0,BREG                        ;
14699          000000                  MICPC=MICPC+1
14700  034350  000400                  .WORD   .$.
14701  034352                          MOVE    BREG,SPAD (16)                  ;FOR RETURN ADDRESS ...
14702          000001                  MICPC=MICPC+1
14703  034352  063236                  .WORD   .$.
14704  034354                          MOVE    BREG,SPAD (1)                   ;
14705          000002                  MICPC=MICPC+1
14706  034354  063221                  .WORD   .$.
14707  034356                          MOVE    BREG,SPAD (2)                   ;
14708          000003                  MICPC=MICPC+1
14709  034356  063222                  .WORD   .$.
14710  034360                          MOVE    BREG,SPAD (3)                   ;
14711          000004                  MICPC=MICPC+1
14712  034360  063223                  .WORD   .$.
14713  034362                          MOVE    # 220,BREG      ;GET READY TO SETUP POWER FAIL..
14714          000005                  MICPC=MICPC+1
14715  034362  000620                  .WORD   .$.
14716  034364                          MOVE    INP1 (CSR0),SPAD (0)    ;GET CNTL/O REGISTER..
14717          000006                  MICPC=MICPC+1
14718  034364  123000                  .WORD   .$.
14719  034366                          OR      BREG,SPAD (0),OUT1 (CSR0)       ;SET RDO, POWER FAIL BIT..
14720          000007                  MICPC=MICPC+1
14721  034366  061300                  .WORD   .$.
14722  034370                          MOVE    # 200,BREG
14723          000010                  MICPC=MICPC+1
14724  034370  000600                  .WORD   .$.
14725  034372                          MOVE    INP1 (CSR11),SPAD (0)   ;GET UPMS REGISTER.
14726          000011                  MICPC=MICPC+1
14727  034372  123220                  .WORD   .$.
14728  034374                          OR      BREG,SPAD (0),OUT1 (CSR11)      ;SET BR+REQ,VCTR:=XX0..
14729          000012                  MICPC=MICPC+1
14730  034374  061311                  .WORD   .$.
14731  034376                   2$:    MOVE    INP1 (CSR11),BREG       ;IS REQ GRANTED??
```

```
14732                000013                          MICPC=MICPC+1
14733  034376 120620                                 .WORD   .$.
14734  034400                                        BR7     2$              ;NO, SPIN ON IT.
14735                000014                          MICPC=MICPC+1
14736  034400 103413                                 .WORD   .$.
14737  034402                               3$:       MOVE    INP1 <CSR0>,BREG        ;IS INTERRUPT SERVICED??
14738                000015                          MICPC=MICPC+1
14739  034402 120400                                 .WORD   .$.
14740  034404                                        BR4     3$              ;NO, WAIT...
14741                000016                          MICPC=MICPC+1
14742  034404 103015                                 .WORD   .$.
14743                                        ;:*
14744                                        ;:*     NOW SET UP FOR FORCE POWER FAIL.
14745                                        ;:*
14746  034406                                        MOVE    # 002,BREG      ;GET READY TO SET FORCE POWER FAIL.
14747                000017                          MICPC=MICPC+1
14748  034406 000402                                 .WORD   .$.
14749  034410                                        MOVE    INP1 <CSR11>,SPAD <0>   ;GET UPMS REGISTER...
14750                000020                          MICPC=MICPC+1
14751  034410 123220                                 .WORD   .$.
14752  034412                                        OR      BREG,SPAD <0>,OUT1 <CSR11>      ;SET AC LO..
14753                000021                          MICPC=MICPC+1
14754  034412 061311                                 .WORD   .$.
14755  034414                               4$:
14756
14757  034414                                        MOVE    INP1 <CSR0>,BREG        ;IS POWER FAIL SERVED??
14758                000022                          MICPC=MICPC+1
14759  034414 120400                                 .WORD   .$.
14760  034416                                        BR7     6$              ;NO, UPDATE DELAY COUNT.
14761                000023                          MICPC=MICPC+1
14762  034416 103425                                 .WORD   .$.
14763  034420                                        SBR     9$              ;YES, SCOPE THE TEST..
14764                000024                          MICPC=MICPC+1
14765  034420 100432                                 .WORD   .$.
14766  034422                               6$:       SINC    SPAD <1>                ;UPDATE DELAY COUNT.
14767                000025                          MICPC=MICPC+1
14768  034422 063061                        .WORD     .$.!.DSP
14769  034424                                        SADC    SPAD <2>                ;UPDATE DELAY COUNT.
14770                000026                          MICPC=MICPC+1
14771  034424 063102                        .WORD     .$.!.DSP
14772  034426                                        SADC    SPAD <3>                ;UPDATE DELAY COUNT.
14773                000027                          MICPC=MICPC+1
14774  034426 063103                        .WORD     .$.!.DSP
14775  034430                                        BC      12$             ;IS IT DELAYED ENOUGH?
14776                000030                          MICPC=MICPC+1
14777  034430 101032                                 .WORD   .$.
14778  034432                                        SBR     4$              ;NO, WAIT
14779                000031                          MICPC=MICPC+1
14780  034432 100422                                 .WORD   .$.
14781  034434                               12$:
14782  034434                               9$:
14783  034434                                        MOVE    INP1 <CSR0>,BREG        ;DUMMY WAIT LOOP
14784                000032                          MICPC=MICPC+1
14785  034434 120400                                 .WORD   .$.
14786  034436                                        SBR     9$              ;DO THE NEXT PASS.
14787                000033                          MICPC=MICPC+1
```

```
14788  034436  100432                       .WORD   .$.
14789  034440                       TSTEN:
14790
14791
14792                               ;SUBROUTINES
14793                               ;-----------
14794
14795
14796                               .EVEN
14797                               ;*************************************************************************
14798                               ;*
14799                               ;*      ERROR REPORT ROUTINE. INTERRUPTS AT LOCATION XX0.
14800                               ;*      1) CSR4:=GOOD DATA      2) CSR5:=BAD DATA
14801                               ;*      3) CSR3:=ERROR TYPE     4) CSR6:=ERROR PC IN MICRO-CODE.
14802                               ;*      5) CSR7:=MISCALLANEOUS INFORMATION.
14803                               ;*
14804                               ;*************************************************************************
14805                               ;
14806  034440                               SBEGIN ,377
14807  034440                               SLOC    :.1000
14808  034440                       ERORD:  SINC    SPAD (16)       ;PREPARE FOR RETURN.
14809          000400                       MICPC=MICPC+1
14810  034440  063076               .WORD   .$.!.DSP
14811  034442                       ERORD1: SINC    SPAD (16)
14812          000401                       MICPC=MICPC+1
14813  034442  063076               .WORD   .$.!.DSP
14814  034444                               MOVE    BREG,OUT1 (CSR6)        ;ERROR PC.
14815          000402                       MICPC=MICPC+1
14816  034444  061226                       .WORD   .$.
14817  034446                               MOVE    BREG,SPAD (17)  ;SAVE RETURN ADDRESS.
14818          000403                       MICPC=MICPC+1
14819  034446  063237                       .WORD   .$.
14820  034450                               MOVE    # 201,BREG      ;SET UP RD+I & ERROR
14821          000404                       MICPC=MICPC+1
14822  034450  000601                       .WORD   .$.
14823  034452                               MOVE    INP1 (CSRD),SPAD (0)    ;GET THE BSELO.
14824          000405                       MICPC=MICPC+1
14825  034452  123000                       .WORD   .$.
14826  034454                               OR      BREG,SPAD (0),OUT1 (CSRD)       ;SET RD+I & ERROR IN SELO.
14827          000406                       MICPC=MICPC+1
14828  034454  061300                       .WORD   .$.
14829  034456                               MOVE    INP1 (CSR11),SPAD (0)   ;GET UPMS REGISTER.
14830          000407                       MICPC=MICPC+1
14831  034456  123220                       .WORD   .$.
14832  034460                               MOVE    # 357,BREG              ;GET THE MASK FOR PGMCLK.
14833          000410                       MICPC=MICPC+1
14834  034460  000757                       .WORD   .$.
14835  034462                               AND     BREG,SPAD (0),SPAD (0)  ;MASK OUT PGMCLK BIT.
14836          000411                       MICPC=MICPC+1
14837  034462  063260                       .WORD   .$.
14838  034464                               MOVE    # 200,BREG              ;PREPARE FOR INTERRUPT.
14839          000412                       MICPC=MICPC+1
14840  034464  000600                       .WORD   .$.
14841  034466                               OR      BREG,SPAD (0),OUT1 (CSR11)      ;SET BR+RD AT XX0.
14842          000413                       MICPC=MICPC+1
14843  034466  061311                       .WORD   .$.
```

```
14844  034470                  51$:  MOVE    INP1 (CSR11),BREG       ;IS BR+RQ GRANTED?
14845         000414                  MICPC=MICPC+1
14846  034470 120620                 .WORD   .$.
14847  034472                        BR7     51$                     ;NO, SPIN ON IT
14850         000415                  MICPC=MICPC+1
14851  034472 107414                 .WORD   .$.
14852  034474                  52$:  MOVE    INP1 (CSR0),BREG        ;IS ERROR INTERRUPT SERVICED?
14853         000416                  MICPC=MICPC+1
14854  034474 120400                 .WORD   .$.
14855  034476                        BB0     52$                     ;NO, SPIN ON IT.
14856         000417                  MICPC=MICPC+1
14857  034476 106016                 .WORD   .$.
14860  034500                        BR7     53$                     ;LOOP ON ERROR?
14861         000420                  MICPC=MICPC+1
14862  034500 107423                 .WORD   .$.
14863  034502                        MOVE    SPAD (16),BREG  ;NO, POP THE RETURN ADDRESS.
14864         000421                  MICPC=MICPC+1
14865  034502 060616                 .WORD   .$.
14866  034504                        SADD    BREG,SPAD (17),SPAD (17)        ;NO, POP THE RETURN ADDRESS.
14867         000422                  MICPC=MICPC+1
14870  034504 063017                 .WORD   .$.
14871  034506                  53$:  MOVE    # 0,BREG                ;
14872         000423                  MICPC=MICPC+1
14873  034506 000400                 .WORD   .$.
14874  034510                        MOVE    BREG,SPAD (16)  ;RESET SPAD (16)...
14875         000424                  MICPC=MICPC+1
14876  034510 063236                 .WORD   .$.
14877  034512                        MOVE    BREG,OUT1 (CSR0)        ;RESET THE TALK REGISTER...
15000         000425                  MICPC=MICPC+1
15001  034512 061220                 .WORD   .$.
15002  034514                        SBR     SPAD (17) PAGE0 ;RETURN TO CALLER.
15003         000426                  MICPC=MICPC+1
15004  034514 160617                 .WORD   .$.
15005
15006                          ;
15007                          ;:*************************************************************************
15010                          ;:*
15011                          ;:*     SCOPE DATA ROUTINE
15012                          ;:*
15013                          ;:*************************************************************************
15014                          ;
15015  034516                  SCP10:  SINC    SPAD (16)       ;PREPARE FOR RETURN.
15016         000427                  MICPC=MICPC+1
15017  034516 063076          .WORD   .$.!.DSP
15020  034520                  SCP110: SINC    SPAD (16)       ;PREPARE FOR RETURN.
15021         000430                  MICPC=MICPC+1
15022  034520 063076          .WORD   .$.!.DSP
15023  034522                        MOVE    BREG,SPAD (17)  ;RETURN ADDRESS.
15024         000431                  MICPC=MICPC+1
15025  034522 063237                 .WORD   .$.
15026  034524                        MOVE    # 202,BREG      ;BREG+RD+I, SCOP1.
15027         000432                  MICPC=MICPC+1
15030  034524 000602                 .WORD   .$.
15031  034526                        MOVE    BREG,OUT1 (CSR0)        ;SET RD+I, SCOP1.
15032         000433                  MICPC=MICPC+1
15033  034526 061220                 .WORD   .$.
15034  034530                        MOVE    INP1 (CSR11),SPAD (0)   ;GET UPMS REGISTER
```

```
14900                   000434          MICPC=MICPC+1
14901 034530  123220                    .WORD   .$.
14902 034532                            MOVE    # 357,BREG              ;GET MASK FOR PGMCLK.
14903                   000435          MICPC=MICPC+1
14904 034532  000757                    .WORD   .$.
14905 034534                            AND     BREG,SPAD <0>,SPAD <0>  ;MASK OUT PGMCLK BIT.
14906                   000436          MICPC=MICPC+1
14907 034534  063260                    .WORD   .$.
14908 034536                            MOVE    # 200,BREG              ;PREPARE FOR INTERRUPT.
14909                   000437          MICPC=MICPC+1
14910 034536  000600                    .WORD   .$.
14911 034540                            OR      BREG,SPAD <0>,OUT1 <CSR11>      ;SET BR+RQ AT XXD.
14912                   000440          MICPC=MICPC+1
14913 034540  061311                    .WORD   .$.
14914 034542                    54$:    MOVE    INP1 <CSR11>,BREG       ;IS BR+RQ GRANTED?
14915                   000441          MICPC=MICPC+1
14916 034542  120620                    .WORD   .$.
14917 034544                            BR7     54$                     ;NO, SPIN ON IT.
14918                   000442          MICPC=MICPC+1
14919 034544  107441                    .WORD   .$.
14920 034546                    55$:    MOVE    INP1 <CSR0>,BREG        ;IS SCOP1 SERVICED?
14921                   000443          MICPC=MICPC+1
14922 034546  120400                    .WORD   .$.
14923 034550                            BR1     55$                     ;NO, SPIN ON IT.
14924                   000444          MICPC=MICPC+1
14925 034550  106443                    .WORD   .$.
14926 034552                            BR7     56$                     ;LOOP ON ERROR?
14927                   000445          MICPC=MICPC+1
14928 034552  107450                    .WORD   .$.
14929 034554                            MOVE    SPAD <16>,BREG  ;NO, POP THE RETURN ADDRESS.
14930                   000446          MICPC=MICPC+1
14931 034554  060616                    .WORD   .$.
14932 034556                            SADD    BREG,SPAD <17>,SPAD <17>        ;NO, POP THE RETURN ADDRESS.
14933                   000447          MICPC=MICPC+1
14934 034556  063017                    .WORD   .$.
14935 034560                    56$:    MOVE    # 0,BREG                ;NO, POP THE RETURN ADDRESS.
14936                   000450          MICPC=MICPC+1
14937 034560  000400                    .WORD   .$.
14938 034562                            MOVE    BREG,SPAD <16>  ;RESET SPAD <16>...
14939                   000451          MICPC=MICPC+1
14940 034562  063236                    .WORD   .$.
14941 034564                            MOVE    BREG,OUT1 <CSR0>        ;RESET THE TALK REGISTER.
14942                   000452          MICPC=MICPC+1
14943 034564  061220                    .WORD   .$.
14944 034566                            SBR     SPAD <17> PAGE0 ;RETURN TO CALLER.
14945                   000453          MICPC=MICPC+1
14946 034566  160617                    .WORD   .$.
14947
14948                           ;
14949                           ;:*************************************************************************
14950                           ;:*
14951                           ;:*     SCOPE THE TEST ROUTINE.
14952                           ;:*
14953                           ;:*************************************************************************
14954 034570                     ;
14955                   000454  SCPE0:  MOVE    BREG,SPAD <17>  ;SAVE RETURN ADDRESS.
                                        MICPC=MICPC+1
```

```
14956  034570  063237                  .WORD   .$.
14957  034572                          MOVE    # 200,BREG      ;RD+0, XN BREG
14958          000455                  MICPC=MICPC+1
14959  034572  000600                  .WORD   .$.
14960  034574                          MOVE    BREG,OUT1 <CSR2>        ;BET RD+0 IN CSR2.
14961          000456                  MICPC=MICPC+1
14962  034574  061222                  .WORD   .$.
14963  034576                          MOVE    INP1 <CSR11>,SPAD <0>   ;GET UPMS REG.
14964          000457                  MICPC=MICPC+1
14965  034576  123220                  .WORD   .$.
14966  034600                          MOVE    # 357,BREG              ;GET MASK FOR PGMCLK.
14967          000460                  MICPC=MICPC+1
14968  034600  000757                  .WORD   .$.
14969  034602                          AND     BREG,SPAD <0>,SPAD <0>  ;MASK OUT PGMCLK BIT.
14970          000461                  MICPC=MICPC+1
14971  034602  063260                  .WORD   .$.
14972  034604                          MOVE    # 300,BREG      ;SET BR RQ & VCTR:=XX4.
14973          000462                  MICPC=MICPC+1
14974  034604  000700                  .WORD   .$.
14975  034606                          OR      BREG,SPAD <0>,OUT1 <CSR11>      ;SET BR+RQ & VCTR:=XX4.
14976          000463                  MICPC=MICPC+1
14977  034606  061311                  .WORD   .$.
14978  034610                  42$:    MOVE    INP1 <CSR11>,BREG       ;IS BR+RQ GRANTED?
14979          000464                  MICPC=MICPC+1
14980  034610  120620                  .WORD   .$.
14981  034612                          BR7     42$             ;NO, SPIN ON IT.
14982          000465                  MICPC=MICPC+1
14983  034612  107464                  .WORD   .$.
14984  034614                          MOVE    # 00,BREG       ;PREPARE TO CLEAR VCTR:=XX4.
14985          000466                  MICPC=MICPC+1
14986  034614  000400                  .WORD   .$.
14987  034616                          MOVE    INP1 <CSR11>,SPAD <0>   ;GET UPMS REG
14988          000467                  MICPC=MICPC+1
14989  034616  123220                  .WORD   .$.
14990  034620                          AND     BREG,SPAD <0>,OUT1 <CSR11>      ;RESET XX4
14991          000470                  MICPC=MICPC+1
14992  034620  061271                  .WORD   .$.
14993  034622                  45$:    MOVE    INP1 <CSR2>,BREG        ;IS INTERRUPT SERVICED.
14994          000471                  MICPC=MICPC+1
14995  034622  120440                  .WORD   .$.
14996  034624                          BR7     45$             ;NO, SPIN ON IT.
14997          000472                  MICPC=MICPC+1
14998  034624  107471                  .WORD   .$.
14999  034626                          SBR     SPAD <17> PAGE0 ;RETURN
15000          000473                  MICPC=MICPC+1
15001  034626  160617                  .WORD   .$.
15002
15003                          ;;
15004                          ;;**************************************************************
15005                          ;;*
15006                          ;;*     MICRO-PROCESSOR PC SEQUENCE ERROR.
15007                          ;;*
15008                          ;;**************************************************************
15009  034630                  ;;
15010          000474          UPCSQR: MOVE    # 24,BREG               ;ERROR CODE.
15011  034630  000424                  MICPC=MICPC+1
                                        .WORD   .$.
```

```
15012  034632                          MOVE    BREG,OUT1 <CSR3>        ;SET ERROR TYPE.
15013         000475                   MICPC=MICPC+1
15014  034632 061223                   .WORD   .$.
15015  034634                          MOVE    # 201,BREG              ;SET UP FOR ERROR REPORT.
15016         000476                   MICPC=MICPC+1
15017  034634 000601                   .WORD   .$.
15018  034636                          MOVE    BREG,OUT1 <CSR0>        ;
15019         000477                   MICPC=MICPC+1
15020  034636 061220                   .WORD   .$.
15021  034640                          MOVE    # 200,BREG              ;BREG=BR-RQ. AT VCTR:=XX0.
15022         000500                   MICPC=MICPC+1
15023  034640 000600                   .WORD   .$.
15024  034642                          MOVE    BREG,OUT1 <CSR11>       ;IS INTERRUPT SERVICED???
15025         000501                   MICPC=MICPC+1
15026  034642 061231                   .WORD   .$.
15027  034644                   1$:    MOVE    INP1 <CSR11>,BREG       ;IS INTERRUPT SERVICED??
15028         000502                   MICPC=MICPC+1
15029  034644 120620                   .WORD   .$.
15030  034646                          BBZ     1$                      ;SPIN ON IT.
15031         000503                   MICPC=MICPC+1
15032  034646 107502                   .WORD   .$.
15033  034650                   2$:    MOVE    INP1 <CSR0>,BREG        ;IS ERROR SERVICED???
15034         000504                   MICPC=MICPC+1
15035  034650 120400                   .WORD   .$.
15036  034652                          BRO     2$                      ;WAIT...
15037         000505                   MICPC=MICPC+1
15038  034652 106104                   .WORD   .$.
15039  034654                          CALL    SCPE0                   ;SCOPE THE FAILING TEST.
15040  034654                          MOVE    # <MICPC+3>,BREG
15041         000506                   MICPC=MICPC+1
15042  034654 000510                   .WORD   .$.
15043  034656                          SBR     SCPE0
15044         000507                   MICPC=MICPC+1
15045  034656 104454                   .WORD   .$.
15046  034660                   3$:    MOVE    INP1 <CSR0>,BREG        ;WAIT FOR PDP11 TO
15047         000510                   MICPC=MICPC+1
15048  034660 120400                   .WORD   .$.
15049  034662                          SBR     3$                      ;.....................
15050         000511                   MICPC=MICPC+1
15051  034662 104510                   .WORD   .$.
15052  034664                   MCSREN:
15053                                   ;;
15054                                   ;;*****************************************************
15055                                   ;;*
15056                                   ;;*     INPUT INTERRUPT SERVICE ROUTINE
15057                                   ;;*     1. TAKES CARE OF ERROR REPORTS.
15058                                   ;;*     2. PROVIDE SERVICE FOR SCOPE DATA.
15059                                   ;;*
15060                                   ;;*****************************************************
15061                                   ;;
15062  034664                   INISR: ;MOV     R0,-(SP)                ;PUSH R0.
15063  034664 013737 001442 035356     MOV     NEXT,SVNXT              ;SAVE NEXT TEST ADDRESS.
15064  034672 013737 001206 035354     MOV     SLPADR,SVLPDR           ;SAVE LOOP ADDRESS.
15065  034700 032711 000001            BIT     #BIT0,(R1)              ;IS ERROR BIT SET?
15066  034704 001012                   BNE     ERR1                    ;YES, SERVE THE ERROR INTERRUPT.
15067  034706 032711 000002            BIT     #BIT1,(R1)              ;IS IT SCOPE DATA SET?
```

```
15068  034712  001075                      BNE     SCPDT           ;YES, GO SERVE IT.
15069  034714  032711  000020              BIT     #BIT4,(R1)      ;IS PWR. FAIL SERVICE SET?
15070  034720  001105                      BNE     PWFAIL          ;GO SERVE POWER FAIL INTERRUPT.
15071                                      ;
15072  034722  104401  011032              TYPE    ,MILINT         ;TYPE ILLEGAL INTERRUPT.
15073  034726  000000                      HALT                    ;WAIT FOR OPERATOR ACTION.
15074  034730  000577                      BR      RETN1           ;CONTINUE....
15075  034732  012737  035022  001442  ERR1:  MOV     #3$,NEXT        ;LOAD RETURN ADDRESS.
15076  034740  012737  035022  001206          MOV     #3$,SLPADR      ;LOAD RETURN ADDRESS.
15077  034746  116137  000004  001224          MOVB    4(R1),$GDDAT    ;LOAD GOOD DATA.
15078  034754  116137  000005  001226          MOVB    5(R1),$BDDAT    ;LOAD BAD DATA.
15079  034762  116137  000006  001222          MOVB    6(R1),$BDADR    ;LOAD UP PC.
15080  034770  042737  000777  035020          BIC     #777,6$         ;CLEAR THE ERROR TYPE FIELD.
15081  034776  156137  000003  035020          BISB    3(R1),6$        ;SET THAT IN ERROR TRAP.
15082  035004  116137  000007  001220          MOVB    7(R1),$GDADR    ;SET CSR FOR REFERENCE.
15083  035012  012737  035354  007070          MOV     #SVLPDR,ERTAB1  ;SAVE ERROR PC...
15084                                      ;
15085  035020  104000              6$:     ERROR                   ;REPORT THE ERROR.
15086  035022  012737  001460  007070  3$:      MOV     #SAVPC,ERTAB1   ;RESTORE ERROR PC POINTER.
15087  035030  032777  000400  144202          BIT     #SW08,@SWR      ;GO TO TOP OF TEST?
15088  035036  001020                          BNE     1$              ;BRANCH IF YES.
15089  035040  032777  002000  144172          BIT     #SW10,@SWR      ;GO TO NEXT TEST?
15090  035046  001412                          BEQ     2$              ;BRANCH IF NO.
15091  035050  013737  035356  001206          MOV     SVNXT,$LPADR    ;SET FOR NEXT TEST.
15092  035056  012706  001200                  MOV     #STACK,SP       ;RESET SP.
15093  035062  012737  000200  177776          MOV     #PR4,PS         ;SET PRIORITY 4 SO KMC CAN INTRRUPT.
15094  035070  000177  144112                  JMP     @$LPADR         ;GO TO SPECIFIED TEST.
15095  035074  042711  000200              2$:     BIC     #BIT7,(R1)      ;CLEAR LOOP ON ERROR BIT.
15096  035100                              1$:     ;MOV    #1,CLKCTR       ;RESET CLOCK.
15097  035100  042711  000001                  BIC     #BIT0,(R1)      ;CLEAR ERROR BIT IN BSELO.
15098  035104  000511                          BR      RETN1           ;RETURN.
15099  035106  005037  001444  SCPDT:  CLR     LOCK            ;CLEAR FLAG
15100  035112  104405                  SCOP1                   ;SCOPE THE DATA.
15101  035114  005737  001444          TST     LOCK            ;IS FLAG SET?-I.E SCOPE THE DATA.
15102  035120  001002                  BNE     SCPDT1          ;YES, GO SET ACCORDINGLY.
15103  035122  042711  000200          BIC     #BIT7,(R1)      ;CLEAR SCOPE DATA LOOP BIT.
15104  035126                      SCPDT1:
15105  035126  042711  000002          BIC     #BIT1,(R1)      ;CLEAR SCOPE DATA BIT.
15106  035132  000476                  BR      RETN1           ;RETURN.
15107                                  ;:*     THIS IS PART OF POWER FAIL TEST
15108                                  ;:*     AFTER KMC CAUSES FORCE POWER FAIL
15109                                  ;:*     AND SETTING UP FOR POWER FAIL THIS
15110                                  ;:*     FUNCTIONS DONE HERE.
15111  035134  012737  035202  001442  PWFAIL: MOV     #30$,NEXT       ;RESTORE.
15112  035142  005037  011112          CLR     TEMP
15113  035146  012737  035202  001206  MOV     #30$,SLPADR     ;RESTORE.
15114  035154  013746  000024          MOV     @#24,-(SP)      ;STORE POWER FAIL ADDRESS.
15115  035160  012737  035204  000024  MOV     #12$,@#24       ;SET UP FOR FORCE POWER FAIL.
15116  035166  042711  000020          BIC     #BIT4,(R1)      ;SIGNAL POWER FAIL SET UP.
15117  035172  005237  011112  15$:     INC     TEMP            ;WAIT FOR POWER FAIL.
15118  035176  001375                  BNE     15$             ;BR IF DELAY NOT DONE.
15119  035200  104017                  ERROR   17              ;ERROR, NO POWER FAIL.
15120  035202  000427          30$:    BR      33$             ;
15121                                  ;:*
15122                                  ;:*     RESTARTS HERE IN CASE OF POWER FAIL.
15123                                  ;:*
```

```
15124  035204  012737  035222  000024  12$:    MOV     #21$,@#24       ;POWER UP ADDRESS.
15125  035212  010637  035220          24$:    MOV     SP,24$          ;STORE STACK.
15126  035216  000000                  21$:    HALT                    ;WAIT FOR POWER UP SEQUENCE.
15127  035220  000000                  24$:    .WORD   0               ;
15128  035222  013706  035220          21$:    MOV     24$,SP          ;RESTORE STACK.
15129  035226  022626                          POP2SP                  ;POP STACK TWICE.
15130  035230  012737  035262  001206          MOV     #33$,SLPADR     ;
15131  035236  012737  035262  001442          MOV     #33$,NEXT       ;SAVE FOR RETURN ON ERROR.
15132  035244  012637  000024                  MOV     (SP)+,@#24      ;RESTORE TRUE POWER FAIL ADDRESS.
15133  035250  022737  007100  000024          CMP     #SPWRDN,@#24    ;IS IT CORRECT?
15134  035256  001401                          BEQ     33$             ;BR IF YES.
15135  035260  104017                          ERROR   17              ;ERROR, STACK IS IN CORRECT.
15136  035262  012737  007100  000024  33$:    MOV     #SPWRDN,@#24    ;RESTORE TRUE POWER FAIL ADDRESS.
15137  035270  012706  001200                  MOV     #STACK,SP       ;RESET STACK...
15138  035274  005711                          TST     (R1)            ;IS KMC STILL RUNNING.E.E.INIT BLOCKED.
15139  035276  100407                          BMI     RETN2           ;YES RETURN.
15140  035300  012737  035316  001206          MOV     #RETN2,SLPADR
15141  035306  012737  035316  001442          MOV     #RETN2,NEXT
15142  035314  104016                          ERROR   16              ;ERROR, BUS INIT WAS NOT BLOCKED.
15143  035316  013737  035356  001206  RETN2:  MOV     SVNXT,SLPADR    ;GO TO NEXT TEST.
15144  035324  000177  143656                  JMP     @SLPADR         ;
15145  035330  012737  000001  036150  RETN1:  MOV     #1,CLKCTR       ;RESET CLOCK.
15146  035336  013737  035356  001442          MOV     SVNXT,NEXT      ;RESTORE:::
15147  035344  013737  035354  001206          MOV     SVLPDR,SLPADR   ;RESTORE...
15148  035352  000002                          RTI
15149  035354  000000          SVLPDR: 0
15150  035356  000000          SVNXT:  0
15151
15152
15153
15154                                  ;::
15155                                  ;::*****************************************************
15156                                  ;::*
15157                                  ;::*     OUTPUT INTERRUPT SERVICE ROUTINE
15158                                  ;::*     1. TAKE CARE OF SCOPE TEST
15159                                  ;::*
15160                                  ;::*****************************************************
15161                                  ;::
15162  035360  132761  000020  000002  OUISR:  BITB    #BIT4,2(R1)     ;IS IT FOR HELL RAISER???
15163  035366  001015                          BNE     HERISR          ;YES DO THAT.
15164  035370  013737  001206  035534          MOV     SLPADR,SVLPAD   ;SAVE LOOP ADDRESS.
15165  035376  012737  035452  001206          MOV     #SKOPE,SLPADR   ;STORE RETURN ADDRESS.
15166  035404  000004                          SCOPE                   ;SCOPE THE TEST.
15167  035406  013737  035534  001206          MOV     SVLPAD,SLPADR   ;NO ITERATION.
15168  035414  005137  036146                  COM     DONE            ;SETUP DONE FLAG
15169  035420  000432                          BR      RET1
15170  035422  142761  000020  000002  HERISR:  BICB    #BIT4,2(R1)     ;YES,ALL SET FOR HELL RAISER.
15171  035430  013761  034312  000004          MOV     HELDAT,4(R1)    ;LOAD CSR.
15172  035436  142761  000002  000002          BICB    #BIT1,2(R1)     ;CLEAR LOAD CSR BIT.
15173  035444  105061  000002                  CLRB    2(R1)           ;CLEAR CSR2.
15174  035450  000416                          BR      RET1
15175  035452  013737  035534  001206  SKOPE:  MOV     SVLPAD,SLPADR   ;RESTORE RETURN ADDRESS.
15176                                  ;       BIC     #BIT7,2(R1)     ;CLEAR RO=0.
15177  035460  122761  000024  000003          CMPB    #24,3(R1)       ;IS IT UPC $Q, ERROR???
15178  035466  001013                          BNE     RET2            ;NO,THEN LET IT BE SCOPED BY
15179                                                                  ;MICRO-PROCESSOR SCOPE ROUTINE.
```

```
15180  035470  105061  000003                  CLRB    3(R1)           ;CLEAN THE WORLD..
15181  035474  013737  001206  001442          MOV     SLPADR,NEXT     ;ELSE, START FROM BEGINING.
15182  035502  005137  036146                  COM     DONE            ;
15183  035506  012737  000001  036150  RET1:   MOV     #1,CLKCTR       ;RESET CLOCK.
15184  035514  000002                          RTI                     ;RETURN TO WAIT CLOCK.
15185  035516  012737  000001  036150  RET2:   MOV     #1,CLKCTR               ;RESET WAIT CLOCK.
15186  035524  142761  000200  000002          BICB    #BIT7,2(R1)     ;CLEAR RD+0.
15187  035532  000002                          RTI                     ;RETURN.
15188  035534  000000                  SVLPAD: 0
15189
15190
15191
15192
15193
15194
15195
15196
15197
15198
15199
15200
15201
15202
15203
15204  035536                          ;:#####################################################################
15205  035536                          ;:#
15206  035536                          ;:#     ROUTINE TO LOAD MICRO-CODE INTO CRAM...
15207  035536                          ;:#     1. LOADS TEST.
15208  035536                          ;:#     2. LOADS UTILITY ROUTINES..
15209  035536                          ;:#     R1 CONTAINS THE CSR AT THE TIME OF ENTRY...
15210  035536                          ;:#
15211  035536                          ;:#####################################################################
15212  035536                          ;
15213  035536                          ;:#########################################
15214  035536                          ;
15215  035536                          LDVRUT:
15216  035536  012700  002000          WCRAM:  ;CLR    R0              ;R0=POINTS TO CRAM ADDRESS.
15217  035542  012711  040000                  MOV     #2000,R0        ;
15218  035546  042711  140000                  MOV     #BIT14,(R1)     ;MASTER CLEAR KMC11
15219  035552  012702  104474                  BIC     #BIT15!BIT14,(R1) ;AND SHUT IT OFF.
15220  035556  005011                          MOV     #104474,R2      ;R2=ADR UPCSDR[MICRO-INSTRUCTION]
15221  035560  010061  000004          6$:     CLR     (R1)            ;START WITH CLEAN WORLD.
15222  035564  010261  000006                  MOV     R0,4(R1)        ;LOAD CRAM ADDRESS.
15223  035570  012711  002000                  MOV     R2,6(R1)        ;LOAD INSTRUCTION.
15224  035574  012711  022000                  MOV     #BIT10,(R1)     ;SET RDWD.
15225  035600  005300                          MOV     #BIT13!BIT10,(R1) ;WRITE IT.
15226  035602  001365                          DEC     R0              ;COUNT BY ONE.
15227  035604  013602                          BNE     6$              ;BR IF NOT DONE.
15228  035606  010237  036152                  MOV     (SP)+,R2        ;
15229  035612  062746  000002                  MOV     R2,TSTRT        ;SAVE START ADDRESS...
15230  035616  013737  001442  036154          ADD     #2,-(SP)                ;POP RETURN ADDRESS...
15231  035624  022737  003662  036154          MOV     NEXT,MCEND      ;SET UP END POINTER.
15232  035632  001003                          CMP     #SEOP,MCEND     ;IS IT LAST TEST?
15233  035634  012737  034440  036154          BNE     1$              ;NO, THEN IT'S O.K.
15234  035642  005011                          MOV     #TSTEN,MCEND    ;SET UP END POINTER.
15235  035646  010061  000004          1$:     CLR     (R1)            ;START WITH CLEAN WORLD.
15236  035652  012261  000006                  MOV     (R2)+,6(R1)     ;LOAD CRAM ADDRESS...
15237  035656  012711  002000                  MOV     #BIT10,(R1)     ;LOAD WORD TO BE WRITTEN.
15238  035662  012711  022000                  MOV     #BIT13!BIT10,(R1) ;WRITE IT...
15239  035666  005200                          INC     R0              ;WRITE IT.
15240  035670  022700  002000                  CMP     #2000,R0        ;UPDATE CRAM ADDRESS.
15241  035674  002403                          BLT     2$              ;OVER FLOW??
15242  035676  023702  036154                  CMP     MCEND,R2        ;YES, THATS IT! NO MORE.
15243  035702  003360                          BGT     1$              ;IS IT DONE?
15244  035704  022737  034664  036154  2$:     CMP     #MCSREN,MCEND   ;NO,CONTINUE LOADING.
15245  035712  001410                          BEQ     3$              ;IS UTILITY ALSO LOADED?
                                                                        ;YES,RETURN.
```

```
 15236  035712  012700  000400            MOV     #400,R0                 ;R0=POINTS CRAM ADDRESS.
 15237  035716  012702  034440            MOV     #ERORD,R2               ;R2:=POINTS TO MICRO-CODE.
 15238  035722  012737  034664  036154    MOV     #MCSREN,MCEND           ;SET UP END POINTER.
 15239  035730  000744                    BR      1$                      ;START LOADING UTILITY.
 15240  035732  005011              3$:   CLR     (R1)                    ;END WITH CLEAN WORLD.
 15241                                    ;
 15242                                    ;;*****************************************************************
 15243                                    ;;*
 15244                                    ;;*     VERIFY THE MICRO-CODE LOADED.
 15245                                    ;;*     VERIFIES THE LOADED TEST MICRO-CODE. AND UTILITY
 15246                                    ;;*     ROUTINES...
 15247                                    ;;*     R1 CONTAINS DEVICE CSR AT THE TIME OF ENTRY...
 15248                                    ;;*
 15249                                    ;;*****************************************************************
 15250                                    ;
 15251                                    ;
 15252                                    ;;*************************************
 15253                                    ;
 15254                                    VERFY:  CLR     R2                      ;R2 GETS CRAM ADDRESS.
 15255  035734  005002                    MOV     TSTRT,R0                ;LOAD START ADDRESS...
 15256  035736  013700  036152            MOV     NEXT,MCEND              ;SET UP END POINTER.
 15257  035742  013737  001442  036154    CMP     #SEOP,MCEND             ;IS IT LAST TEST??
 15258  035750  022737  003662  036154    BNE     1$                      ;NO,THEN IT'S O.K.
 15259  035756  001003                    MOV     #TSTEN,MCEND            ;SET UP END POINTER.
 15260  035760  012737  034440  036154    CLR     (R1)                    ;CLEAR SELD.
 15261  035766  005011              1$:   MOV     R2,4(R1)                ;SET THE CRAM ADDRESS.
 15262  035770  010261  000004            MOV     #BIT10,(R1)             ;SET ROMD.
 15263  035774  012711  002000            CMP     6(R1),(R0)+             ;CHECK IF RIGHT?
 15264  036000  026120  000006            BEQ     3$                      ;CHECK IF DONE?
 15265  036004  001403                    TYPE    ,MLDER                  ;LOADING ERROR.
 15266  036006  104401  036156            HALT                            ;WAIT FOR OPERATOR.
 15267  036012  000000
 15268  036014  005202              3$:   INC     R2                      ;BUMP CRAM ADDRESS.
 15269  036016  023700  036154            CMP     MCEND,R0                ;IS IT DONE??
 15270  036022  003361                    BGT     1$                      ;NO,CHECK NEXT.
 15271  036024  022737  034664  036154    CMP     #MCSREN,MCEND           ;IS IT DONE???
 15272  036032  001410                    BEQ     6$                      ;YES,RETURN...
 15273  036034  012702  000400            MOV     #400,R2                 ;R2 CONTAINS CRAM ADDRESS...
 15274  036040  012737  034664  036154    MOV     #MCSREN,MCEND           ;SET UP END POINTER...
 15275  036046  012700  034440            MOV     #ERORD,R0               ;R0=POINTS TO MICRO-CODE.
 15276  036052  000745                    BR      1$                      ;START CHECKING UTILITIES.
 15277  036054  005011              6$:   CLR     (R1)                    ;END WITH CLEAN WORLD.
 15278  036056  012711  040000            MOV     #BIT14,(R1)             ;MASTER CLEAR THE KMC.
 15279  036062  042711  140000            BIC     #BIT14!BIT15,(R1)               ;
 15280                                    ;
 15281                                    ;
 15282                                    ;;*****************************************************************
 15283                                    ;;*
 15284                                    ;;*     WAIT CLOCK :- WAITS FOR KMC TO RESPOND
 15285                                    ;;*     BACK WITH DONE OR ERROR IN SPECIFIC TIME IF DOES NOT
 15286                                    ;;*     GET ONE GENERATES A TIME OUT ERROR.
 15287                                    ;;*
 15288                                    ;;*****************************************************************
 15289                                    ;
 15290  036066  005037  036150    WATCLK: CLR     CLKCTR                  ;INITIALIZE CLOCK.
 15291  036072  005037  036146            CLR     DONE                    ;WAIT.
```

```
015282 036076 012711 100000          MOV     #BIT15,(R1)     ;INITIATE THE TEST...
015283 036102 005737 036146    1$:   TST     DONE            ;IS DONE FLAG SET?
015284 036106 001010                 BNE     3$              ;YES, START THE NEXT TEST.
015285 036110 005237 036150          INC     CLKCTR          ;COUNT BY ONE.
015286 036114 005337 036150          DEC     CLKCTR          ;WAIT CLOCK.
015287 036120 005237 036150          INC     CLKCTR          ;  "   "
015288 036124 001366                 BNE     1$              ;GO CHECK AGAIN. IF NOT TIMED OUT.
015289 036126 000207                 RTS     PC              ;
015300 036130 013737 001442 001206 3$:   MOV     NEXT,@LPADR     ;SET THE NEXT TEST ADDRESS.
015301 036136 012706 001200          MOV     #STACK,SP       ;POP THE STACK.
015302 036142 000177 143040          JMP     @#LPADR         ;START THE NEXT TEST.
015303 036146 000000                 DONE:   0
015304 036150 000000                 CLKCTR: 0
015305 036152 000000                 TSTRT:  0
015306 036154 000000                 MCEND:  0
015307 036156 046200 040517 044504   MLDER:  .ASCIZ  <200>/LOADING ERROR/
015308 036164 043516 042440 051122
015309 036172 051117 000
015310       036176                  .EVEN
015311 036176                        $FINI
015312 036176 044600 052502 027523   EM1:    .ASCIZ  <200>↑IBUS/IBUS# REGISTER DATA TEST↑
       036236 200 041111 051525      EM2:    .ASCIZ  <200>↑IBUS/IBUS# REGISTER DUAL ADDRESSING TEST↑
       036307 200 051102 051040      EM3:    .ASCIZ  <200>/BR REGISTER DATA TEST/
       036336 051600 051103 052101   EM4:    .ASCIZ  <200>/SCRATCH PAD DATA TEST/
       036365 200 041523 040522      EM5:    .ASCIZ  <200>/SCRATCH PAD DUAL ADDRESSING TEST/
       036427 200 040515 047111      EM6:    .ASCIZ  <200>/MAIN MEMORY DATA TEST/
       036456 046600 044501 020116   EM7:    .ASCIZ  <200>/MAIN MEMORY DUAL ADDRESSING TEST/
       036520 040600 052125 020117   EM10:   .ASCIZ  <200>/AUTO INCRINC FUNCTION TEST/
       036553 200 050116 020122      EM11:   .ASCIZ  <200>/NPR TEST/
       036605 200 052515 052114      EM12:   .ASCIZ  <200>/MULTIPLE NPR TEST/
       036610 047200 047117 042440   EM13:   .ASCIZ  <200>/NON EX MEM FAILED TO SET/
       036642 050200 047522 051107   EM14:   .ASCIZ  <200>/PROGRAM CLOCK TEST/
       036666 040600 052514 043040   EM15:   .ASCIZ  <200>/ALU FUNCTION WITH C BIT CLEAR TEST/
       036732 050200 052517 051105   EM16:   .ASCIZ  <200>/POWER FAIL; BUS INIT HAS NOT BLOCKED/
       037000 043200 051117 042503   EM17:   .ASCIZ  <200>/FORCE POWER FAIL ERROR/
       037030 047200 044517 042523   EM20:   .ASCIZ  <200>/NOISE TEST ON IBUS#,IBUS,SPAD,MEMORY/
       037076 040600 052514 041440   EM21:   .ASCIZ  <200>/ALU C BIT TEST FAILURE/
       037126 052200 046511 020105   EM22:   .ASCIZ  <200>/TIME OUT ERROR/
       037146 040600 052514 043040   EM23:   .ASCIZ  <200>/ALU FUNCTION TEST WITH C BIT SET/
       037210 052600 041520 051440   EM24:   .ASCIZ  <200>/UPC SEQUENCE ERROR/
       037234 044200 046105 020114   EM25:   .ASCIZ  <200>/HELL RAISER TEST/

       037256 043600 047517 020104   DH1:    .ASCIZ  <200>/GOOD    BAD    UPC    REGISTER/
       037317 200 047507 042117      DH2:    .ASCIZ  <200>/GOOD    BAD/
       037334 043600 047517 020104   DH3:    .ASCIZ  <200>/GOOD    BAD    UPC    ADDRESS/
       037373 200 047507 042117      DH4:    .ASCIZ  <200>/GOOD    BAD    UPC    FUNCTION/
              037434                 .EVEN

       037434 000004                 DT1:    4
       037436 003 006                        .BYTE   3,6
       037440 001224                 $GDDAT
       037442 003 004                        .BYTE   3,4
       037444 001226                 $BDDAT
       037446 003 006                        .BYTE   3,6
       037450 001222                 $BDADR
       037452 003 006                        .BYTE   3,6
```

```
037454  001220                          $GDADR
037456  000002                  DT2:    2
037460     003      004                 .BYTE   3,4
037462  001224                          $GDDAT
037464     003      004                 .BYTE   3,4
037466  001226                          $GDDAT
037470                          CORMAX:
        000001                          .END
```

```
ABASE = 000000            298      339
ACDW1 = 000000            298      341
ACDW2 = 000000            298      342
ACPUDP= 000000            298      313
ADDW0 = 000000            298      343
ADDW1 = 000000            298      344
ADDW10= 000000            298      353
ADDW11= 000000            298      354
ADDW12= 000000            298      355
ADDW13= 000000            298      356
ADDW14= 000000            298      357
ADDW15= 000000            298      358
ADDW2 = 000000            298      345
ADDW3 = 000000            298      346
ADDW4 = 000000            298      347
ADDW5 = 000000            298      348
ADDW6 = 000000            298      349
ADDW7 = 000000            298      350
ADDW8 = 000000            298      351
ADDW9 = 000000            298      352
ADEVCT= 000000            298      304
ADEVM = 000000            298      340
ADRCNT  006047           1400*    1415*    1424#
ADVANC= 104420           1569#
AENV  = 000002              1#      298      309
AENVM = 000000            298      310
AFATAL= 000000            298      301
AMADR1= 000000            298      326
AMADR2= 000000            298      330
AMADR3= 000000            298      333
AMADR4= 000000            298      336
AMAMS1= 000000            298      320
AMAMS2= 000000            298      328
AMAMS3= 000000            298      331
AMAMS4= 000000            298      334
AMSCNO= 000000            298      306
AMSGLG= 000000            298      307
AMSGTY= 000000            298      300
AMTYP1= 000000            298      321
AMTYP2= 000000            298      329
AMTYP3= 000000            298      332
AMTYP4= 000000            298      335
APASS = 000000            298      303
APRIOR= 000000            298
APTCSU= 000040           1125     1230#
APTENV= 000001           1118     1186     1228#    1630
APTSIZ= 000200           1227#
APTSPC= 000100           1120     1188     1229#
APT.SI  012552            796     2210#
ASWREG= 000000            298      311
ATESTM= 000000            298      302
AUDONE  003354            833      854      893#
AUNIT = 000000            298      305
AUSTRT  003126            832#
AUSWR = 000000            298      312
AUTO.S  012152            794     1954#
```

```
AVECT1= 000000  298  337
AVECT2= 000000  298  338
BASE    013770  2221* 2232  2246* 2256#
BINWRD  006402  1472* 1475* 1476  1513#
BIT0  = 000001  161#  1737  1738  15065  15097
BIT00 = 000001  151#  161
BIT01 = 000002  150#  160
BIT02 = 000004  149#  159
BIT03 = 000010  148#  158
BIT04 = 000020  147#  157
BIT05 = 000040  146#  156
BIT06 = 000100  145#  155
BIT07 = 000200  144#  154
BIT08 = 000400  143#  153
BIT09 = 001000  142#  152
BIT1  = 000002  160#  801  1731  1737  1738  2091  15067  15105  15172
BIT10 = 002000  141#  2072  15213  15214  15227  15228  15263
BIT11 = 004000  140#  1064
BIT12 = 010000  139#  2026  2089
BIT13 = 020000  138#  2029  2075  2093  15214  15228
BIT14 = 040000  137#  1046  2040  2042  2093  2104  15207  15208  15278  15279
BIT15 = 100000  136#  741  15208  15279  15292
BIT2  = 000004  159#  801  942  1738
BIT3  = 000010  158#  2095  2102
BIT4  = 000020  157#  1721  1744  1746  15069  15116  15162  15170
BIT5  = 000040  156#  2207
BIT6  = 000100  155#  1726  1727  2097
BIT7  = 000200  154#  1727  1846  2207  15095  15103  15186
BIT8  = 000400  153#  2084  2086  2099  2101  2107  2110  2167
BIT9  = 001000  152#  2081  2107  2110  2165  2167
BM      010603  1769#  2053
BPTVEC= 000014  168#
BRCR  = 000011  1#
BRLVL   013504  2162  2172  2180  2193#
BRW     004354  947  1080#
CHRCNT  006400  1470* 1473  1477  1493* 1511#  1512
CKSWR   011216  772  1086  1576  1794#
CKSWR1  011272  1806#  1818
CKSWR2  011304  1809#
CKSWR3  011310  1811#
CKSWR4  011314  1812#  1820  1827
CKSWR5  011420  1795  1797  1801  1836#
CLKCTR  036150  15145* 15183* 15185* 15290* 15295* 15296* 15297* 15304#
CLKX    001452  373#
CMPRE   034034  14057  14067  14077  14087  14097  14107  14117  14129  14139  14149  14159  14169  14179
                14189  14199  14215  14234  14247  14260  14273  14286  14299  14312  14325  14338  14416#
CNERR   011001  866  1769#
CNT.MA  002302  387  621#  740  742  744  1874
CNVRT = 104417  863  972  974  976  978  1568#  1611  1613  1688  1809
CONERR  010734  861  1769#
CONN    010643  1769#  2031
CONTAB  003330  870  878#
CONVRT= 104416  813  869  1567#  1627  1976
CORMAX  037470  15312#
CR    = 000015  76#  1164  1174
CREAM   001502  386#  739*  1870*  1871  1873*  1877
```

```
CRLF   = 000200          77#  1135  1174
CSR    = 110335        1769#  1980
CSRMAP  012154         1956#
CSR0   = 000000           1#  13876  14049  14717  14720  14738  14758  14784  14824  14827  14851  14872  14897
                       14921  14942  15019  15034  15047
CSR1   = 000001           1#
CSR10  = 000010           1#   7362   7404   7407   7548   7608   7611   7614   7638   7641   7794   7821   7922
                        7982   7985   8021  14565  14568
CSR11  = 000011           1#   7551   7554   7812   7815   7824   7858   7861  13729  13754  13757  13783  14532
                       14544  14553  14726  14729  14732  14750  14753  14830  14842  14845  14900  14912  14915
                       14964  14976  14979  14988  14991  15025  15028
CSR2   = 000002           1#  13879  14059  14536  14556  14574  14580  14961  14984
CSR3   = 000003           1#   2321   2401   2510   2598   2715   2795   2912   2992   3119   3208   3345   3443
                        3560   3640   3757   3837   3954   4034   4151   4231   4348   4428   4545   4625   4742
                        4822   4939   5019   5133   5203   5319   5404   5526   5611   5733   5818   5940   6025
                        6147   6232   6354   6438   6561   6646   6768   6853   6975   7060   7182   7267   7437
                        7471   7674   7714   7842   8072   8191   8348   8501   8591   8751   8878   9193   9510
                        9827  10144  10461  10778  11095  11412  11729  12046  12363  12680  12997  13314  13631
                       13738  13770  13808  13886  14076  14376  14444  14460  15013
CSR4   = 000004           1#   2312   2392   2504   2584   2706   2786   2903   2983   3110   3199   3336   3434
                        3551   3631   3748   3828   3945   4025   4142   4222   4339   4419   4536   4616   4733
                        4813   4930   5010   5127   5197   5307   5392   5514   5599   5721   5806   5928   6013
                        6135   6220   6342   6427   6519   6634   6720   6841   6963   7048   7170   7255   7431
                        7465   7682   7702   8060   8102   8339   8495   8565   8745   9184   9501   9818  10135
                       10452  10769  11086  11403  11720  12037  12354  12671  12988  13305  13622  13885  14079
                       14367  14438  14490  14508  14595  14638  14644
CSR5   = 000005           1#   2315   2395   2512   2592   2709   2789   2906   2986   3113   3202   3339   3437
                        3554   3634   3751   3831   3948   4028   4145   4225   4342   4422   4539   4619   4736
                        4816   4933   5013   5130   5200   5313   5398   5520   5605   5727   5812   5934   6019
                        6141   6226   6348   6433   6555   6640   6762   6847   6969   7054   7176   7261   7425
                        7462   7688   7708   8066   8185   8342   8489   8579   8739   9187   9504   9821  10138
                       10455  10772  11089  11406  11723  12040  12357  12674  12991  13308  13625  13888  14089
                       14370  14432  14493  14563  14598
CSR6   = 000006           1#  13842  14099  14815
CSR7   = 000007           1#   2327   2407   2524   2604   2721   2801   2918   2998   3125   3214   3351   3449
                        3566   3646   3763   3843   3960   4040   4157   4237   4354   4434   4551   4631   4748
                        4828   4945   5025   5228   5410   5532   5617   5739   5824   5946   6031   6153   6238
                        6360   6445   6567   6652   6774   6859   6981   7066   7188   7273   8197   8354   8507
                        9128   9199   9445   9516   9762   9833  10079  10150  10396  10467  10713  10784  11030
                       11101  11347  11418  11664  11735  11981  12052  12298  12369  12615  12686  12932  13003
                       13249  13320  13566  13637  13895  14109
CYCLE   011464          948   1009   1861#
DATABP  006754         1600*  1603   1625   1628#
DATACL= 104413         1564#
DATAHD  006742         1599*  1621   1624#
DATIN  = 000001           1#
DATIO  = 000300           1#
DATI1  = 000001           1#
DATOUT= 000021            1#
DATO0  = 000002           1#
DATO1  = 000003           1#
DDISP  = 177570          83#   239#   268
DELAY  = 104411        1562#
DEVTAB  003342          836   883#
DH1     037256          418   421   427   430  15312#
DH2     037317          424   442   445   463   478  15312#
```

```
DH3     037334          433     436     439   15312#
DH4     037373          454     472   15312#
DISPLA  001242          268#    758*    764*   1076*
DISPRE  000174          229#    764
DONE    036146        15168*  15182*  15291*  15293   15303#
DSWR  = 177570           82#    238#    267
DT1     037434          419     422     428     431     434     437     440     455     473   15312#
DT2     037456          425     443     446     464     479   15312#
DZDME = ******  U       414
DZDMG = ******  U       414
DZKCA = 000000         1078    1089    1903    2275    2472    2669    2866    3064    3281    3514    3711    3908    4105
                       4302    4499    4696    4893    5091    5268    5475    5682    5889    6096    6303    6510    6717
                       6924    7131    7337    7511    7754    7891    8131    8288    8449    8662    8828    8954    9271
                       9588    9905   10222   10539   10856   11173   11490   11807   12124   12441   12758   13075   13392
                      13707   13850   14474   14692
EMTVEC= 000030          171#
EM1     036176          417   15312#
EM10    036520          438   15312#
EM11    036553          441   15312#
EM12    036565          444   15312#
EM13    036610          447   15312#
EM14    036642          450   15312#
EM15    036666          453   15312#
EM16    036732          456   15312#
EM17    037000          459   15312#
EM2     036235          420   15312#
EM20    037030          462   15312#
EM21    037076          465   15312#
EM22    037126          468   15312#
EM23    037146          471   15312#
EM24    037210          474   15312#
EM25    037234          477   15312#
EM3     036307          423   15312#
EM4     036336          426   15312#
EM5     036365          429   15312#
EM6     036427          432   15312#
EM7     036456          435   15312#
ERCT00  002304          623#
ERCT01  002310          626#
ERCT02  002314          629#
ERCT03  002320          632#
ERCT04  002324          635#
ERCT05  002330          638#
ERCT06  002334          641#
ERCT07  002340          644#
ERCT10  002344          647#
ERCT11  002350          650#
ERCT12  002354          653#
ERCT13  002360          656#
ERCT14  002364          659#
ERCT15  002370          662#
ERCT16  002374          665#
ERCT17  002400          668#
EROR  = 035350         2281#   2335    2415    2478#   2532    2612    2675#   2729    2809    2872#   2926    3006    3070#
                       3133    3222    3287#   3359    3457    3520#   3574    3654    3717#   3771    3851    3914#   3968
                       4048    4111#   4165    4245    4308#   4362    4442    4505#   4559    4639    4702#   4756    4836
```

```
                    4899#  4953   5033   5097#  5274#  5333   5418   5481#  5540   5625   5688#  5747   5832
                    5955#  5954   6039   6102#  6161   6246   6309#  6368   6453   6516#  6575   6660   6723#
                    6782   6867   6930#  6989   7074   7137#  7196   7281   7343#  7517#  7760#  7897#  8137#
                    8205   8294#  8362   8455#  8668#  8759   8834#  8960#  9136   9207   9277#  9453   9524
                    9594#  9770   9841   9911#  10087  10158  10228#  10404  10475  10545#  10721  10792  10862#
                   11038  11109  11179#  11355  11426  11496#  11672  11743  11813#  11989  12060  12130#  12306
                   12377  12447#  12623  12694  12764#  12940  13011  13081#  13257  13328  13398#  13574  13645
                   13713#  13856#  14480#  14698#
ERORD   034440     14808#  15237  15275
ERORD1  034442     14811#
ERR1 =  035352      2281#  2478#  2675#  2872#  3070#  3287#  3520#  3717#  3914#  4111#  4308#  4505#  4702#
                    4899#  5097#  5141   5211   5274#  5481#  5688#  5895#  6102#  6309#  6516#  6723#  6930#
                    7137#  7343#  7445   7479   7517#  7682   7722   7760#  7850   7897#  8080   8137#  8294#
                    8455#  8515   8599   8668#  8834#  8896   8960#  9277#  9594#  9911#  10228#  10545#  10862#
                   11179#  11496#  11813#  12130#  12447#  12764#  13081#  13398#  13713#  13746  13778  13816  13856#
                   14384  14452  14480#  14658  14698#
ERR11= 001000         2#  2281   2478   2675   2872   3070   3287   3520   3717   3914   4111   4308   4505
                    4702   4899   5097   5274   5481   5688   5895   6102   6309   6516   6723   6930   7137
                    7343   7517   7760   7897   8137   8294   8455   8668   8834   8960   9277   9594   9911
                   10228  10545  10862  11179  11496  11813  12130  12447  12764  13081  13398  13713  13856
                   14480  14698
ERR     003244      845    858#   862
ERRMSG  006730     1598*   1616   1619#
ERRPC   003322      864    875#
ERRVEC= 000004      164#   1051   1052*  1054*  1057*
ERR1    034732    15066  15075#
ERTAB0  007064     1613   1653#
ERTAB1  007070     1655#  15083*  15086*
EXIT =  000205      191#
EXITER  007036     1639   1644#
FLOAT   003156      838#   844
FRCPFW= 000002        1#
FY      003202      846#   850    855
GNS  =  ****** U   1552   1555   1556   1557   1558   1559   1560   1561   1562   1563   1564   1565   1566
                   1567   1568   1569
HALTS   006760     1583   1630#
HELDAT  034312    14511  14517  14672#  15171
HERISR  035422    15163  15170#
HILIM   006042    1397*  1405   1421#
HT   =  000011      74#  1133   1174
IBADO =  000004       1#
IBAD1 =  000005       1#
IBA16 =  000004       1#
IBA17 =  000010       1#
ICR   =  000012       1#
INCHAR  011424     1812   1840#
INIFLG  001506      391#   767    797    804*
INISR   034664     1903  15062#
INLP1   005756     1402#  1411
INNPR =  000000       1#
INPUT =  134415    1566#  1916   1967   1979   1987   2044   2052
INTTY   013520     1997   2016   2032   2201#
IOTVEC= 000020      169#
KMACTV  001470      380#   781    906*   907   1861   1875   1962*  2122*  2128*  2129*  2133   2158   2224*
                   2225
KMCM    011022      872   1769#
```

```
KMCR00  002100  536#
KMCR01  002110  541#
KMCR02  002120  546#
KMCR03  002130  551#
KMCR04  002140  556#
KMCR05  002150  561#
KMCR06  002160  566#
KMCR07  002170  571#
KMCR10  002200  576#
KMCR11  002210  581#
KMCR12  002220  586#
KMCR13  002230  591#
KMCR14  002240  596#
KMCR15  002250  601#
KMCR16  002260  606#
KMCR17  002270  611#
KMCSR   002066  519#   832*   847    853*   882    1012   1078   1691   1745   1879*  1888   1942
KMCSRH  002070  520#  1726*  1727*  1731*  1737*  1738*  1744*  1746*  1888*  1889*  1890
KMCTL   002072  521#  1890*  1891*  1892
KMNUM   001472  381#   735    909   1960*  1974*  2114*  2115   2123   2125
KMPO4   002074  522#  1715*  1721   1758   1763   1892*  1893*  1894
KMPO6   002076  523#  1732*  1894*  1895*
KMRLVL  002060  516#   982    983*   1897*  1898*  1899   1904*
KMRVEC  002056  515#   982*  1015   1880*  1881*  1897   1903*
KMS100  002102  537#
KMS101  002112  542#
KMS102  002122  547#
KMS103  002132  552#
KMS104  002142  557#
KMS105  002152  562#
KMS106  002162  567#
KMS107  002172  572#
KMS110  002202  577#
KMS111  002212  582#
KMS112  002222  587#
KMS113  002232  592#
KMS114  002242  597#
KMS115  002252  602#
KMS116  002262  607#
KMS117  002272  612#
KMS200  002104  538#
KMS201  002114  543#
KMS202  002124  548#
KMS203  002134  553#
KMS204  002144  558#
KMS205  002154  563#
KMS206  002164  568#
KMS207  002174  573#
KMS210  002204  578#
KMS211  002214  583#
KMS212  002224  588#
KMS213  002234  593#
KMS214  002244  598#
KMS215  002254  603#
KMS216  002264  608#
KMS217  002274  613#
```

```
KMS300  002106  539#
KMS301  002116  544#
KMS304  002126  549#
KMS305  002136  554#
KMS306  002146  559#
KMS307  002156  564#
KMS310  002166  569#
KMS311  002176  574#
KMS312  002206  579#
KMS313  002216  584#
KMS314  002226  589#
KMS315  002236  594#
KMS316  002246  599#
KMS317  002256  604#
KMTLVL  002064  518#    984     985*    1901*   1902*   1906*
KMTVEC  002062  517#    984*    1899*   1900*   1901    1905*
KM.END  002300  616#  1958    2229
KM.MAP  002100  386     535#    739     806     816     1871    1873    1956    1961    2150    2227    2231
LDATI = 000010    1#
LDATO = 000010    1#
LDVRWT  035536  2275    2472    2669    2866    3064    3281    3514    3711    3908    4105    4302    4499    4696
                4893    5091    5268    5475    5682    5889    6096    6303    6510    6717    6924    7131    7337
                7511    7754    7891    8131    8288    8449    8662    8828    8954    9271    9588    9905    10222
                10539   10856   11173   11490   11807   12124   12441   12758   13075   13392   13707   13850   14474
                14692   15204#
LF    = 000012    75#   1168    1174
LINE    010545  1769#   2045
LOBITS  006046  1399*   1423#
LOCK    001444    367#  1089*   1430*   1607    15099*  15101
LOKFLG  001510    393#
LOLIM   006040  1396*   1407    1420#
LULOOP= 000040    1#
MAINT = 000017    1#
MARHLD= 000000    1#
MARINC= 014000    1#    7946    7991    7994    8690    8727    8988    8991    8994    8997    9000    9003    9006
                9009    9012    9015    9018    9021    9024    9027    9030    9033    9036    9039    9042    9045
                9048    9051    9054    9057    9060    9063    9066    9069    9072    9075    9078    9081    9098
                9101    9160    9169    9172    9231    9305    9308    9311    9314    9317    9320    9323    9326
                9329    9332    9335    9338    9341    9345    9347    9350    9353    9356    9359    9362    9365
                9368    9371    9374    9377    9380    9383    9386    9389    9392    9395    9398    9415    9418
                9477    9486    9489    9548    9622    9625    9628    9631    9634    9637    9640    9643    9646
                9649    9652    9655    9658    9661    9664    9667    9670    9673    9676    9679    9682    9685
                9688    9691    9694    9697    9700    9703    9706    9709    9712    9715    9732    9735    9794
                9803    9806    9865    9939    9942    9945    9948    9951    9954    9957    9960    9963    9966
                9969    9972    9975    9978    9981    9984    9987    9990    9993    9996    9999    10002   10005
                10008   10011   10014   10017   10020   10023   10026   10029   10032   10049   10052   10111   10120
                10123   10182   10256   10259   10262   10265   10268   10271   10274   10277   10280   10283   10286
                10289   10292   10295   10298   10301   10304   10307   10310   10313   10316   10319   10322   10325
                10328   10331   10334   10337   10340   10343   10346   10349   10366   10369   10428   10437   10440
                10499   10573   10576   10579   10582   10585   10588   10591   10594   10597   10600   10603   10606
                10609   10612   10615   10618   10621   10624   10627   10630   10633   10636   10639   10642   10645
                10648   10651   10654   10657   10660   10663   10666   10683   10686   10745   10754   10757   10816
                10890   10893   10896   10899   10902   10905   10908   10911   10914   10917   10920   10923   10926
                10929   10932   10935   10938   10941   10944   10947   10950   10953   10956   10959   10962   10965
```

```
                    10968  10971  10974  10977  10980  10983  11000  11003  11062  11071  11074  11133  11207
                    11210  11213  11216  11219  11222  11225  11228  11231  11234  11237  11240  11243  11246
                    11249  11252  11255  11258  11261  11264  11267  11270  11273  11276  11279  11282  11285
                    11288  11291  11294  11297  11300  11317  11320  11379  11388  11391  11450  11524  11527
                    11530  11533  11536  11539  11542  11545  11548  11551  11554  11557  11560  11563  11566
                    11569  11572  11575  11578  11581  11584  11587  11590  11593  11596  11599  11602  11605
                    11608  11611  11614  11617  11634  11637  11646  11705  11708  11767  11841  11844  11847
                    11850  11853  11856  11859  11862  11865  11868  11871  11874  11877  11880  11883  11886
                    11889  11892  11895  11898  11901  11904  11907  11910  11913  11916  11919  11922  11925
                    11928  11931  11934  11951  11954  12013  12022  12025  12084  12158  12161  12164  12167
                    12170  12173  12176  12179  12182  12186  12188  12191  12194  12197  12200  12203  12206
                    12209  12212  12215  12218  12221  12224  12227  12230  12233  12236  12239  12242  12245
                    12248  12251  12254  12271  12330  12339  12342  12401  12475  12478  12481  12484  12487
                    12490  12493  12496  12499  12502  12505  12508  12511  12514  12517  12520  12523  12526
                    12529  12532  12535  12538  12541  12544  12547  12550  12553  12556  12559  12562  12565
                    12568  12571  12647  12656  12659  12718  12782  12785  12788  12801  12804  12807
                    12810  12813  12816  12819  12822  12825  12828  12831  12834  12837  12840  12843  12846
                    12849  12852  12855  12858  12861  12864  12867  12870  12873  12876  12879  12882  12885
                    12902  12905  12944  12973  12976  13035  13109  13112  13115  13118  13121  13124  13127
                    13130  13133  13136  13139  13142  13145  13148  13151  13154  13157  13160  13163  13166
                    13169  13172  13175  13178  13181  13184  13187  13190  13193  13196  13199  13202  13219
                    13222  13281  13290  13293  13352  13426  13429  13432  13435  13438  13441  13444  13447
                    13450  13453  13456  13459  13462  13465  13468  13471  13474  13477  13480  13483  13486
                    13489  13492  13495  13498  13501  13504  13507  13510  13513  13516  13519  13536  13539
MASKX   001454      13598  13607  13610  13669  13962  14355
                      374#
MASTEK  007767       1609   1769#
MCEND   036154      15220*  15221  15223*  15232  15234  15238*  15257*  15258  15260*  15269  15271  15274*  15306#
MCSREN  034664      15052#  15234  15238  15271  15274
MCSRX   007717        971   1769#
MCT1    014030       2276   2280#
MCT10   015730       3712   3716#
MCT11   016134       3909   3913#
MCT12   016340       4106   4110#
MCT13   016544       4303   4307#
MCT14   016750       4500   4504#
MCT15   017154       4697   4701#
MCT16   017360       4894   4898#
MCT17   017564       5092   5096#
MCT2    014234       2473   2477#
MCT20   017754       5269   5273#
MCT21   020170       5476   5480#
MCT22   020404       5683   5687#
MCT23   020620       5890   5894#
MCT24   021034       6097   6101#
MCT25   021250       6304   6308#
MCT26   021464       6511   6515#
MCT27   021700       6718   6722#
MCT3    014440       2670   2674#
MCT30   022114       6925   6929#
MCT31   022330       7132   7136#
MCT32   022544       7338   7342#
MCT33   022740       7512   7516#
MCT34   023210       7755   7759#
MCT35   023356       7892   7896#
MCT36   023642       8132   8136#
```

```
MCT37   024020          8289    8293#
MCT4    014544          2867    2871#
MCT40   024200          8450    8454#
MCT41   024416          8663    8667#
MCT42   024600          8829    8833#
MCT43   024734          8955    8959#
MCT44   025062          9272    9276#
MCT45   025210          9589    9593#
MCT46   025136          9906    9910#
MCT47   025464         10223   10227#
MCT5    015050          3065    3069#
MCT50   027012         10540   10544#
MCT51   027340         10857   10861#
MCT52   027666         11174   11178#
MCT53   030214         11491   11495#
MCT54   030542         11808   11812#
MCT55   031070         12125   12129#
MCT56   031416         12442   12446#
MCT57   031744         12759   12763#
MCT6    015270          3382    3386#
MCT60   032272         13076   13080#
MCT61   032620         13393   13397#
MCT62   033146         13708   13712#
MCT63   033314         13851   13855#
MCT64   034120         14475   14479#
MCT65   034350         14693   14697#
MCT7    015524          3515    3519#
MDATA   011154          1491    1501    1786#
MEMLIM  001466           379#    930*
MEPASS  007572           970    1769#
MERRPC  010044          1612    1769#
MERRX   007744           977    1769#
MERR2   007617          1769#   2135
MERR3   007644           903    1769#
MICPC = 000511          2281#   2331    2344    2411    2424    2447    2478#   2528    2541    2608    2621    2644    2675#
                        2765    2738    2805    2818    2841    2872#   2922    2935    3002    3015    3038    3070#   3129
                        3142    3218    3231    3254    3287#   3355    3408    3457    3466    3489    3520#   3570    3583
                        3650    3663    3686    3717#   3767    3780    3847    3860    3883    3914#   3964    3977    4044
                        4057    4080    4111#   4161    4174    4241    4254    4277    4308#   4358    4371    4438    4451
                        4474    4505#   4555    4568    4635    4648    4671    4702#   4752    4765    4832    4845    4868
                        4899#   4949    4962    5029    5042    5055    5077#   5137    5147    5207    5217    5243    5274#
                        5329    5342    5414    5427    5450    5481#   5534    5549    5621    5634    5657    5680#   5743
                        5756    5828    5841    5864    5886#   5960    6023    6036    6048    6071    6102#   6157    6170
                        6242    6255    6278    6309#   6364    6377    6449    6462    6485    6516#   6571    6584    6656
                        6669    6682    6723#   6778    6791    6863    6876    6889    6930#   6985    6998    7070    7083
                        7106    7137#   7192    7205    7277    7290    7313    7343#   7441    7475    7485    7517#   7678
                        7718    7728    7760#   7846    7865    7897#   8076    8104    8137#   8201    8214    8264    8294#
                        8358    8371    8424    8455#   8511    8521    8596    8605    8637    8668#   8755    8768    8803
                        8834#   8882    8892    8926    8940#   9132    9145    9203    9216    9243    9277#   9449    9462
                        9520    9533    9560    9594#   9766    9779    9837    9850    9877    9911#  10083   10096   10154
                       10167   10194   10228#  10400   10413   10471   10484   10511   10545#  10717   10730   10788   10801
                       10828   10862#  11034   11047   11105   11118   11145   11179#  11351   11364   11422   11435   11462
                       11496#  11668   11681   11739   11752   11779   11813#  11965   11998   12056   12069   12096   12130#
                       12302   12315   12373   12386   12413   12447#  12619   12632   12690   12703   12730   12764#  12936
                       12949   13007   13020   13047   13081#  13253   13266   13324   13337   13364   13398#  13570   13583
                       13641   13654   13681   13713#  13742   13774   13812   13822   13856#  14053   14063   14073   14083
```

```
                    14093  14103  14113  14125  14135  14145  14155  14165  14175  14185  14195  14211  14230
                    14243  14256  14269  14282  14295  14308  14321  14334  14380  14408  14448  14480#  14654
                    14664  14698#  14807#  15041
MILINT  011032       1769#  15072
MILK    001504        387#   740*    979   1869*  1874*  1878
MLDER   036156      15266  15307#
MLOCK   007670        944   1769#
MLRLD = 010000         1#   7940   7961   8036   8045   8092   8148   8254   8305   8414   8462   8531   8555
                     8615   8675   8715   8761   8774   8841   8961   9038   9138   9151   9157   9209   9222
                     9278   9405   9455   9468   9474   9526   9539   9595   9722   9772   9785   9791   9843
                     9856   9912  10039  10069  10102  10108  10160  10173  10229  10356  10406  10419  10425
                    10477  10490  10546  10673  10723  10736  10742  10794  10807  10863  10940  11040  11053
                    11059  11111  11124  11180  11307  11357  11370  11376  11428  11441  11497  11624  11674
                    11687  11693  11745  11758  11814  11941  11991  12004  12010  12062  12075  12131  12258
                    12308  12321  12327  12379  12392  12448  12473  12486  12538  12544  12636  12709  12765
                    12842  12942  12955  12961  13013  13026  13082  13209  13259  13272  13278  13330  13343
                    13399  13526  13576  13589  13595  13647  13660  13870  14343  14484
MMEM    007771       898   1769#
MMPDT   023572      7925   7931   8112#
MODU    010433      1769#  2015
MPASSX  007733       975   1769#
MPFAIL  007534      1687   1704   1769#
MPRLD = 004000         1#   7943   7964   8039   8048   8151   8257   8308   8417   8465   8534   8558   8618
                     8678   8718   8844   8964   9281   9598   9915  10232  10549  10866  11183  11500  11817
                    12134  12451  12768  13085  13402  13873  14346  14487
MR      007614       951   1769#  1932
MRESET= 004000       191#
MSTCLR= 104410      1561#  1693
MTITLE  001000       237#   771
MTSTN   007755      1610   1769#  1917
MVECX   007725       973   1769#
NEXT    001442       366#  1429   2273*  2279   2470*  2476   2667*  2673   2864*  2870   3062*  3068   3279*
                    3285   3512*  3518   3709*  3715   3906*  3912   4103*  4109   4300*  4306   4497*  4503
                    4694*  4700   4891*  4897   5089*  5095   5286*  5292   5473*  5479   5680*  5686   5887*
                    5893   6094*  6100   6301*  6307   6508*  6514   6715*  6721   6922*  6928   7129*  7135
                    7336*  7341   7509*  7515   7752*  7758   7959*  7965   8129*  8135   8286*  8292   8447*
                    8453   8660*  8666   8826*  8832   8952*  8958   9269*  9275   9586*  9592   9903*  9909
                   10220*  10226  10537*  10543  10854*  10860  11171*  11177  11488*  11494  11805*  11811  12122*
                   12128   12439*  12445  12756*  12762  13073*  13079  13390*  13396  13705*  13711  13848*  13854
                   14472*  14478  14690*  14696  15063  15075*  15111*  15131*  15141*  15146*  15181*  15220  15257
                   15300
NOACT   010703       733   1769#  1863
NODEV   003240       834    856#
NPDAT   022702      7377   7383   7493#
NPRBYT= 000200         1#
NPRC    = 000010       1#
NPRDTI  023152      7569   7575   7736#
NUM     010275      1769#  1968
NXMEM   = 000001       1#
OBRDO   = 000006       1#
OBRD1   = 000007       1#
OBA16   = 000004       1#
OBA17   = 000010       1#
OCR     = 000011       1#
OK      003220       848    851#   874
ONE     001464       378#
```

```
OTNPR = 000020            1#
QUISR   035360         1905   15162#
PACT00  002302          622#
PACT01  002306          625#
PACT02  002312          628#
PACT03  002316          631#
PACT04  002322          634#
PACT05  002326          637#
PACT06  002332          640#
PACT07  002336          643#
PACT10  002342          646#
PACT11  002346          649#
PACT12  002352          652#
PACT13  002356          655#
PACT14  002362          658#
PACT15  002366          661#
PACT16  002372          664#
PACT17  002376          667#
PAGE0 = 000000            1#   14456   14876   14946   15001
PAGE1 = 001000            1#
PAGE2 = 002000            1#
PAGE3 = 003000            1#
PARBIT= 040000          191#
PCLK  = 000020            1#
PENFOR= 004537          191#
PFTAB   007276         1688    1710#
PIRQ  = 177772           81#
PIRQVE= 000240          175#
POPRO = 012600          186#   1643
POP1SP= 005726          184#
POP2SP= 022626          188#  15129
PRIO    010374         1769#   1996
PRIRTY  013772         2217*  2223*  2236   2257#
PR0   = 000000            98#   2193   2194
PR1   = 000040            99#
PR2   = 000100           100#
PR3   = 000140           101#
PR4   = 000200           102#   1907   2195  15093
PR5   = 000240           103#   1904   1906   2196
PR6   = 000300           104#   2197
PR7   = 000340           105#    210    212    214    216   2198
PS    = 177776            78#     79    732*   941*  1907*  2162*  2172*  15093*
PSW   = 177776            79#
PUSHRO= 010046          185#   1640
PUSH1S= 005746          183#
PUSH2S= 024646          187#
PWFAIL  035134         15070   15111#
PWRVEC= 000024          170#   1666*  1667*  1676*  1682*  1701*  1702*
QV.FLG  001511          394#    738*   988*
RABO  = 000200            1#
RACT  = 000100            1#
RBCC  = 000001            1#
RBR   = 000200            1#
RDCHR = 104402         1282   1555#
RDLIN = 104403         1355   1556#
RDOCT = 104404         1404   1557#
```

```
REQM  = 000002  1#
RESREG  000756  1629  1629#
RESVEC= 000010  165#
RESUS = 104407  1560#  1623
RETN1   035330  15074  15098  15106  15145#
RETN2   035316  15139  15140  15141  15143#
RET1    035506  15169  15174  15183#
RET2    035516  15178  15185#
ROMCLK= 104412  1563#  1716  1719  1756  1761
ROMPR = 000001  1#
RQDY  = 000020  1#
RUN     001500  384#  741*  1867*  1868*  1875
SAVACT  001474  382#  901  2133*  2225*
SAVNUM  001476  383#  735*  986*  989*  2126*  2219*
SAVPC   001460  376#  862*  877  1436*  1655  15086
SAVSP   001456  375#
SAVUS = 104406  1559#  1588
SCOPI = 104405  1558#  15100
SCPDT   035106  15068  15099#
SCPDT1  035126  15102  15104#
SCPE  = 035500  2281#  2451  2478#  2648  2675#  2845  2872#  3042  3070#  3258  3287#  3493  3520#
                3690  3717#  3887  3914#  4084  4111#  4281  4308#  4478  4505#  4675  4702#  4872
                4899#  5069  5097#  5267  5274#  5454  5481#  5661  5688#  5858  5895#  6075  6102#
                6282  6309#  6489  6516#  6696  6723#  6903  6930#  7110  7137#  7317  7343#  7489
                7517#  7732  7760#  7939  7897#  8108  8137#  8268  8294#  8428  8455#  8641  8668#
                8807  8834#  8930  8960#  9247  9277#  9564  9594#  9881  9911#  10198  10228#  10515
                10545#  10832  10862#  11149  11179#  11466  11496#  11783  11813#  12100  12130#  12417  12447#
                12734  12764#  13051  13081#  13368  13398#  13685  13713#  13826  13856#  14412  14480#  14668
                14698#
SCPE0   034570  14954#  15045
SCPE1 = 001130  2#  2281  2478  2675  2872  3070  3287  3520  3717  3914  4111  4308  4505
                4702  4899  5097  5274  5481  5688  5895  6102  6309  6516  6723  6930  7137
                7343  7517  7760  7897  8137  8294  8455  8668  8834  8960  9277  9594  9911
                10228  10545  10862  11179  11496  11813  12130  12447  12764  13081  13398  13713  13856
                14480  14698
SCP1  = 035426  2281#  2478  2478#  2675  2675#  2872  2872#  3070  3070#  3287  3287#  3520  3520#
                3146  3235  3287#  3372  3470  3520#  3587  3667  3717#  3784  3864  3914#  3981
                4061  4111#  4178  4259  4308#  4375  4455  4505#  4572  4652  4702#  4769  4849
                4899#  4966  5046  5097#  5274#  5346  5431  5481#  5553  5638  5688#  5760  5845
                5935#  5967  6052  6102#  6174  6259  6309#  6381  6466  6516#  6588  6673  6723#
                6795  6880  6930#  7002  7087  7137#  7209  7294  7343#  7517#  7760#  7897#  8137#
                8218  8294#  8375  8455#  8668#  8772  8834#  8960#  9149  9220  9277#  9466  9537
                9594#  9783  9854  9911#  10100  10171  10228#  10417  10488  10545#  10734  10805  10862#
                11051  11122  11179#  11368  11439  11496#  11685  11756  11813#  12002  12073  12130#  12319
                12390  12447#  12636  12707  12764#  12953  13024  13081#  13270  13341  13398#  13587  13658
                13713#  13856#  14480#  14698#
SCP10   034516  14884#
SCP11 = 035430  2281#  2478#  2675#  2872#  3070#  3287#  3520#  3717#  3914#  4111#  4308#  4505#  4702#
                4899#  5097#  5151  5221  5274#  5481#  5688#  5895#  6102#  6309#  6516#  6723#  6930#
                7137#  7343#  7517#  7760#  7897#  8137#  8294#  8455#  8525  8609  8668#  8834#  8896
                8960#  9277#  9594#  9911#  10228#  10545#  10862#  11179#  11496#  11813#  12130#  12447#  12764#
                13081#  13398#  13713#  13856#  14480#  14698#
SCP110  034520  14887#
SCP111= 001056  2#  2281  2478  2675  2872  3070  3287  3520  3717  3914  4111  4308  4505
                4702  4899  5097  5274  5481  5688  5895  6102  6309  6516  6723  6930  7137
                7343  7517  7760  7897  8137  8294  8455  8668  8834  8960  9277  9594  9911
```

```
                  10220  10545  10862  11179  11496  11813  12130  12447  12764  13081  13398  13713  13856
                  14480  14698
SKOPE   035452    15165  15175#
SOFTSW  011456     1810   1849#
SPACNT= 006401     1471#  1495   1498#  1512#
STACK = 001200       69#   733    920   2278   2475   2672   2869   3067   3284   3517   3714   3911   4108
                   4305   4502   4699   4896   5094   5271   5478   5685   5892   6099   6306   6513   6720
                   6927   7134   7340   7514   7757   7954   8134   8291   8452   8665   8831   8957   9274
                   9591   9808  10225  10542  10859  11176  11493  11810  12127  12444  12761  13078  13395
                  13710  13853  14477  14695  15092  15137  15301
STAT    001450      372#
STAT1   002050      508#  1882#
STAT2   002052      509#  1883#
STAT3   002054      510#  1884#
STKLMT= 177774       80#
STRTSW  001446      371#   773#  776#   777    779    793#   799    801    896    942    949   1910   1933#
                   1963   2146
SVLPAD  035534    15164#  15167  15175  15188#
SVLPDR  035354    15064#  15083  15147  15149#
SVNXT   035356    15063#  15091  15143  15146  15150#
SVOS    006070     1440#
SWFLG   011422      736#  1805#  1832#  1838#
SWR     001240      267#   757#   759    763#   773    901    906   1046   1064   1087   1577   1582   1638
                   1645   1647   1674   1694#  1734   1794   1831#  15087  15089
SWREG   000176      230#   752#   763    1794   1851
SW0   = 000001      133#
SW00  = 000001      123#   133    777    793   1963   2146
SW01  = 000002      122#   132    949   1910   1933
SW02  = 000004      121#   131
SW03  = 000010      120#   130    896
SW04  = 000020      119#   129
SW05  = 000040      118#   128
SW06  = 000100      117#   127   1734
SW07  = 000200      116#   126
SW08  = 000400      115#   125   1645  15087
SW09  = 001000      114#   124   1087
SW1   = 000002        1#   132#
SW10  = 002000      113#  1647  15089
SW11  = 004000      112#
SW12  = 010000      111#  1577
SW13  = 020000      110#  1582
SW14  = 040000      109#
SW15  = 100000      108#
SW2   = 000004        1#   131#
SW3   = 000010      130#
SW4   = 000020      129#
SW5   = 000040      128#
SW6   = 000100      127#
SW7   = 000200      126#
SW8   = 000400      125#
SW9   = 001000      124#
TBITVE= 000014      166#
TEMP    011112     1478   1742#  1747#  1753#  1765#  1784#  15112#  15117#
TIMER = 104414     1565#
TKVEC = 000060      173#
TLAST = 034314     1936  15312#
```

```
TPVEC = 000064          174#
TRAPVE= 000034          172#
TRTVEC= 000014          167#
TSTEN   034440        14789#  15223   15260
TSTRT   036152        15218*  15256   15305#
TST1    013774         1040    1923    1941   2272#
TST10   015674         3512    3708#
TST11   016100         3709    3905#
TST12   016304         3906    4102#
TST13   016510         4103    4299#
TST14   016714         4300    4496#
TST15   017120         4497    4693#
TST16   017324         4694    4890#
TST17   017530         4891    5088#
TST2    014200         2273    2469#
TST20   017720         5089    5265#
TST21   020134         5266    5472#
TST22   020350         5473    5679#
TST23   020564         5680    5886#
TST24   021000         5887    6093#
TST25   021214         6094    6300#
TST26   021430         6301    6507#
TST27   021644         6508    6714#
TST3    014404         2470    2666#
TST30   022060         6715    6921#
TST31   022274         6922    7128#
TST32   022510         7129    7334#
TST33   022704         7335    7508#
TST34   023154         7509    7751#
TST35   023322         7752    7888#
TST36   023606         7889    8128#
TST37   023764         8129    8265#
TST4    014610         2667    2863#
TST40   024144         8266    8446#
TST41   024362         8447    8659#
TST42   024544         8660    8825#
TST43   024700         8826    8951#
TST44   025226         8952    9240#
TST45   025554         9241    9585#
TST46   026102         9586    9902#
TST47   026430         9903   10219#
TST5    015014         2864    3061#
TST50   026756        10220   10536#
TST51   027304        10537   10853#
TST52   027632        10854   11170#
TST53   030160        11171   11487#
TST54   030506        11488   11804#
TST55   031034        11805   12121#
TST56   031362        12122   12438#
TST57   031710        12439   12755#
TST6    015234         3062    3278#
TST60   032236        12756   13072#
TST61   032564        13073   13389#
TST62   033112        13390   13704#
TST63   033260        13705   13847#
TST64   034064        13848   14471#
```

```
TST65   034314          14472  14689#  15312
TST66 = ###### U        14690
TST7    015470           3279   3511#
TTST    004146            945#   947#  1042#
TWOSYN= 010000            191#
TYPDAT  006744           1604   1622   1625#
TYPE  = 104401            771    783    805#    860    865    871    898    903    944    951    970    971    973
                          975    977   1138#   1289   1295   1300   1304   1309   1310   1312   1315   1319   1384
                         1386   1402   1409   1410   1461   1501   1552#  1605   1606   1609   1610   1612   1614
                         1618   1623   1687   1703   1808   1811   1863   1915   1932   1938   1975   1995   2009
                         2014   2023   2030   2037   2135  15072  15266
TYPMSG  006644           1602   1605#
UPCSAR  034630          15009#
VEC     010353           1769#  1988
VECMAP  013234           2134   2146#
VECTR   013766           2216#  2222#   2241    2247#   2255#
VEC4  = 000100              1#
VERFY   035734          15255#
WATCLK  036066          15290#
WCRAM   035536          15205#
WHAT    005760           1395#  1403#
WHERE   006044           1398#  1412   1422#
WHICH   013226           1977   2142#
WRDCNT  006376           1469#  1502#  1510#
WRKO.F  006732           1617   1620#
XABO  = 000200              1#
XACT  = 000100              1#
XBX     006532           1578   1580   1582#
XCSR    004104            972   1010#
XEOM  = 000002              1#
XERR    004126            978   1019#
XHEAD   010051            805   1769#
XPASS   004120            976   1016#
XRDY  = 000020              1#
XSFU  = 000040              1#
XSOM  = 000001              1#
XSTATQ  011064            814   1769#
XTSTN   007072           1611   1656#
XVEC    004112            974   1013#
ZERO    001462            377#
$APTHD  002034            490    496#
$ASTAT= ###### U         1208   1223
$ATYC   004712           1179   1181#
$ATY1   004666           1177#
$ATY3   004674           1123   1178#
$ATY4   004704           1180#  1633
$AUTOB  001234            264#
$BASE   001372            339#  2221
$BDADR  001222            259#  15079#  15312
$BDDAT  001226            261#  15078#  15312
$CDA1   001376            341#  2218
$CDA2   001400            342#
$CHARC  004662           1140#  1150#   1157    1166#   1171#
$CKSUM= ###### U         1555
$CMTAG  001200            247#
$CM1  = 000006            279#   280#    281#    282#    283#    284#    285#
```

```
SCH2  = 000014  279#  280#  281#  282#  283#  284#  285#
SCH3  = 000006  277#  279
SCH4  = 000005  285#  286#  287#  288#  289#  290#
SCNTLG  005524  1330#
SCNTLU  005517  1304  1329#
SCPUOP  001344  313#
SCRAP  = 177777  1#  2260#  2263  2267#  2457#  2460  2464#  2654#  2657  2661#  2851#  2854  2858#
                 3048#  3051  3056#  3264#  3267  3273#  3499#  3502  3506#  3696#  3699  3703#  3893#
                 3896  3900#  4090#  4093  4097#  4287#  4290  4294#  4484#  4487  4491#  4681#  4684
                 4688#  4878#  4881  4885#  5075#  5078  5083#  5253#  5256  5260#  5460#  5463  5467#
                 5667#  5670  5674#  5874#  5877  5881#  6081#  6084  6088#  6288#  6291  6295#  6495#
                 6498  6502#  6702#  6705  6709#  6909#  6912  6916#  7116#  7119  7123#  7323#  7326
                 7329#  7496#  7499  7503#  7739#  7742  7746#  7875#  7878  7883#  8117#  8120  8123#
                 8274#  8277  8280#  8434#  8437  8441#  8647#  8650  8654#  8813#  8816  8820#  8936#
                 8939  8946#  8953#  8956  9263#  9570#  9573  9580#  9807#  9890  9897#  10204#  10207
                 10214#  10521#  10524  10531#  10838#  10841  10848#  11155#  11158  11165#  11472#  11475  11482#
                 11789#  11792  11799#  12106#  12109  12116#  12423#  12426  12433#  12740#  12743  12750#  13057#
                 13060  13067#  13374#  13377  13384#  13691#  13694  13699#  13832#  13835  13842#  14459#  14462
                 14466#  14475#  14678  14683#
SCRLF  001313  292#  1139  1174  1309  1329  1389  1410  1461  1605  1606  1614  1915  1975
SDDM0  001402  343#  2226
SDDM1  001404  344#
SDDM10  001426  353#
SDDM11  001430  354#
SDDM12  001432  355#
SDDM13  001434  356#
SDDM14  001436  357#
SDDM15  001440  358#
SDDM2  001406  345#
SDDM3  001410  346#
SDDM4  001412  347#
SDDM5  001414  348#
SDDM6  001416  349#
SDDM7  001420  350#
SDDM8  001422  351#
SDDM9  001424  352#
SDEVCT  001326  304#
SDEVM  001374  340#  2224
SDOAGN  004100  987  996  1001  1007#
SENDAD  004070  223  769  1003#  1636
SENDCT  004054  998#
SENV  001336  309#  750  786  788  1118  1186  1210  1630  1796
SENVM  001337  310#  791  1120  1125  1188
SEOP  003662  966#  14690  15221  15250
SEOPCT  004046  995#  999
SERFLG  001203  250#  737*  969*  1031  1060  1062*  1082  1587*  1601  1615*  1689*
SERMAX  001215  256#  1082
SERMOR  006502  213  1576#
SERRPC  001216  257#  746*  990*  1039*  1584  1586*  1690*
SERRTB  001512  411#  1597
SERTTL  001212  254#  981  1021  1644*  1886*
SETABL  001336  308#
SETEND  001442  361#  502
SFATAL  001320  301#  1214*
SFFLG  005132  1177*  1180*  1208  1217*  1225#
SFILLC  001256  275#  1143  1174
```

```
$FILLS  001255          274#  1174
$GOOOR  001220          258#  15082*  15312
$GOOAT  001224          260#  15077*  15312
$GET42  004060         1000#
$GTSWR= ******  U       1554
$HD     = 000000          42
$HIBTS  002034          497#
$HIOCT  005712         1377*  1388#
$ICNT   001204          251#  1068*  1069  1071*  1081
$ILLUP  007270         1666   1682   1706#
$INPUT  005714         1393#  1566
$INTAG  001235          265#
$ITEMB  001214          255#  1592*  1632
$LF     001314          293#  1174   1319  1329  1389
$LFLG   005131         1218*  1224#
$LPADR  001206          252#   948*   948*  952   1075*  1077  1081  1931*  1941*  1943  15064  15076*  15091*
                       15094  15113*  15130*  15140*  15143*  15144  15147*  15164  15165*  15167*  15175*  15181  15300*
                       15302
$LPERR  001210          253#
$MADR1  001350          326#
$MADR2  001354          330#
$MADR3  001360          333#
$MADR4  001364          336#
$MAIL   001316          293#   498   502  1074  1118
$MMS1   001346          320#  2220
$MMS2   001352          328#
$MMS3   001356          331#
$MMS4   001362          334#
$MBADR  002036          498#
$MFLG   005130         1178*  1184  1219*  1223#
$MMEM   005542         1333#  1811
$MSGAD  001332          306#  1194*  1197
$MSGLG  001334          307#  1199*
$MSGTY  001316          300#  1192  1200*  1212  1216*
$MSWR   005531         1331#  1808
$MTYP1  001347          321#
$MTYP2  001353          329#
$MTYP3  001357          332#
$MTYP4  001363          335#
$MXCNT  004356         1072   1081#
$N      = 000065           1#  2260  2267  2270  2275  2276  2280  2281#  2457  2464  2467  2472  2473
                       2477   2478#  2654  2661  2664  2669  2670  2674  2675#  2851  2858  2861  2866
                       2867   2871   2872#  3048  3056  3059  3064  3065  3069  3070#  3264  3273  3276
                       3281   3282   3286  3287#  3499  3506  3509  3514  3515  3519  3520#  3696  3703
                       3706   3711   3712  3716  3717#  3893  3900  3903  3908  3909  3913  3914#  4090
                       4097   4100   4105  4106  4110  4111#  4287  4294  4297  4302  4303  4307  4308#
                       4484   4491   4494  4499  4500  4504  4505#  4681  4688  4691  4696  4697  4701
                       4702#  4878   4885  4888  4893  4894  4898  4899#  5075  5083  5086  5091  5092
                       5096   5097#  5253  5260  5263  5268  5269  5273  5274#  5460  5467  5470  5475
                       5476   5480   5481#  5667  5674  5677  5682  5683  5687  5688#  5874  5881  5884
                       5889   5890   5894  5895#  6081  6088  6091  6096  6097  6101  6102#  6288  6295
                       6298   6303   6304  6308  6309#  6495  6502  6505  6510  6511  6515  6516#  6702
                       6709   6712   6717  6718  6722  6723#  6909  6916  6919  6924  6925  6929  6930#
                       7116   7123   7126  7131  7132  7136  7137#  7323  7329  7332  7337  7338  7342
                       7343#  7496   7503  7506  7511  7512  7516  7517#  7739  7746  7749  7754  7755
                       7759   7760#  7875  7883  7886  7891  7892  7896  7897#  8117  8123  8126  8131
```

```
                    8132   8136   8137#  8274   8280   8283   8288   8289   8293   8294#  8434   8441   8444
                    8449   8450   8454   8455#  8647   8654   8657   8662   8663   8667   8668#  8813   8820
                    8823   8828   8829   8833   8834#  8936   8946   8949   8954   8955   8959#  8960#  9253
                    9263   9266   9271   9272   9276   9277#  9570   9580   9583   9588   9589   9593   9594#
                    9897   9899   9900   9905   9906   9910   9911#  10204  10214  10217  10222  10223  10227
                   10228#  10521  10531  10534  10539  10540  10544  10545#  10838  10848  10851  10856  10857
                   10861   10862#  11155  11165  11168  11173  11174  11178  11179#  11472  11482  11485  11490
                   11491   11495  11496#  11769  11799  11802  11807  11808  11812  11813#  12106  12116  12119
                   12124   12125  12129  12130#  12423  12433  12436  12441  12442  12446  12447#  12740  12750
                   12753   12758  12759  12763  12764#  13057  13067  13070  13075  13076  13080  13081#  13374
                   13384   13387  13392  13393  13397  13398#  13691  13699  13702  13707  13708  13712  13713#
                   13832   13842  13845  13850  13851  13855  13856#  14459  14466  14469  14474  14475  14478
                   14480#  14675  14683  14687  14692  14693  14697  14698#  15312#
SNULL   001254      273#  1145   1174
SOVER   004324     1044   1047   1058   1070   1076#
SPASS   001324      303#   968#   980    992#   993#   1010   1018   1066   1082   1885#
SPASTM  002042      500#
SPWRDN  007100      211    734   1666#  1701   15133  15136
SPWRMG  007264     1704#
SPWRUP  007152     1676   1682#
SQUES   001312      291#  1174   1312   1329   1386   1389   1409   1938   2010   2024   2038
SRDCHR  005134     1249#  1555
SRDDEC= ****** U   1558
SRDLIN  005254     1277#  1556
SRDOCT  005554     1350#  1557
SRDSZ = 000007     1270#
SREGAD  001260      277#
SREGO   001262      279#  1445#  1450
SREG1   001264      280#   868#   880    1444#  1451
SREG2   001266      281#  1443#  1452
SREG3   001270      282#  1442#  1453
SREG4   001272      283#  1441#  1454
SREG5   001274      284#  1440#  1455
SRTNAD  004102     1009#
SR2A  = ****** U   1558
SS    = 000067        1#  2273   2281#  2470   2478#  2667   2675#  2864   2872#  3062   3070#  3279   3287#
                    3512   3520#  3709   3717#  3906   3914#  4103   4111#  4300   4308#  4497   4505#  4694
                    4702#  4891   4899#  5089   5097#  5286   5294#  5473   5481#  5680   5688#  5887   5895#
                    6094   6102#  6301   6309#  6508   6516#  6715   6723#  6922   6930#  7129   7137#  7335
                    7343#  7509   7517#  7752   7760#  7889   7897#  8129   8137#  8286   8294#  8447   8455#
                    8660   8668#  8826   8834#  8952   8960#  9269   9277#  9586   9594#  9903   9911#  10220
                   10228#  10537  10545#  10854  10862#  11171  11179#  11488  11496#  11805  11813#  12122  12130#
                   12439   12447#  12756  12764#  13073  13081#  13390  13398#  13705  13713#  13848  13856#  14472
                   14480#  14690  14698#
SSAVRE= ****** U   1558
SSAVR6  007274     1675*  1683   1684*  1685*  1708#
SSCOPE  004134      209   1038#  2190
SSETUP= 000000      991   1039   1238   1335
SSVLAD  004316     1055   1073#
SSVPC = 000040      221#   226
SSWR  = 164000        1#    42    290    291    963    991   1002   1008   1010   1032   1033   1034   1035
                    1046   1058   1060   1061   1062   1063   1064   1076   1081   1705
SSWREG  001340      311#   752
SSWRMK= 000000     1035
STESTN  001322      302#  1074*
STIMES  001310      290#   991*  1063*  1069   1072*  1081
```

```
STKB    001246          270#  1045  1236  1253  1259  1798  1800  1842  2203
STKS    001244          269#  1043  1236  1251  1257  1840  2201
STMPO   001276          285#   807*  1771  2158*  2159*
STMP1   001300          286#   808*  1773
STMP2   001302          287#   810*   831*   858   867*  1775  1971  1974  2061*
STMP3   001304          288#   811*  1777  1983  1986  1991  1994  2048  2051  2056  2059
STMP4   001306          289#   812*  1779  1966*  1978*  2144
STN   = 000001            1#    42
STPB    001255          272#  1163*  1174  1581*  1845*  2206*
STPFLG  001257          276#  1112  1174
STPS    001250          271#  1161  1174  1579  1843  2204
STRAP   006404          215   1529#
STRAP2  006426         1540#  1551
STRP  = 000021         1544#  1553#  1555  1556#  1557#  1558#  1559#  1560#  1561#  1562#  1563#  1564#  1565#
                       1566#  1567#  1568#  1569#  1570#
STRPAD  006440         1534   1551#
STSTM   002040          499#
STSTNM  001202          249#   747*  1031  1073*  1074  1076  1082  1658  1712  1920  1927  1929  2272*
                       2469*  2666*  2863*  3061*  3278*  3511*  3708*  3905*  4102*  4299*  4496*  4693*  4890*
                       5088*  5265*  5472*  5679*  5886*  6093*  6300*  6507*  6714*  6921*  7128*  7334*  7508*
                       7751*  7888*  8128*  8285*  8446*  8659*  8825*  8951*  9268*  9585*  9902*  10219*  10536*
                      10853*  11170*  11487*  11804*  12121*  12438*  12755*  13072*  13389*  13704*  13847*  14471*  14689*
STTYIN  005510         1279   1280  1292  1310  1324  1328#
STYPBN= ****** U       1553
STYPOS= ****** U       1553
STYPE   004404         1112#  1205  1544  1552
STYPEC  004616         1142   1149  1156  1161#  1162
STYPEX  004664         1167   1169  1172#
STYPOC= ****** U       1553
SUNIT   001330          305#
SUNITM  002044          501#
SUSWR   001342          312#
SVECT1  001366          337#  2222  2223
SVECT2  001370          338#
SXTSTR  004174         1041   1049#
SY    = 000000            1#   503#
SSGET4= 000000         1002#
SSINIT= 033440         2281#  2310  2335  2341  2348  2354  2363  2366  2390  2415  2421  2428  2434
                       2443   2451  2454  2478#  2507  2532  2538  2551  2560  2563  2587  2612
                       2618   2625  2631  2640  2648  2651  2675#  2704  2729  2735  2742  2748  2757
                       2760   2784  2809  2815  2822  2828  2837  2845  2848  2872#  2901  2926  2932
                       2939   2945  2954  2957  2981  3006  3012  3019  3025  3034  3042  3045  3070#
                       3108   3133  3139  3146  3152  3161  3164  3197  3222  3228  3236  3241  3250
                       3258   3261  3287#  3334  3359  3365  3372  3378  3387  3390  3432  3457  3463
                       3470   3476  3485  3493  3496  3520#  3549  3574  3580  3587  3593  3602  3605
                       3629   3654  3660  3667  3673  3682  3690  3693  3717#  3746  3771  3777  3784
                       3790   3799  3802  3826  3851  3857  3864  3870  3879  3887  3890  3914#  3943
                       3968   3974  3981  3987  3996  4023  4054  4061  4067  4076  4084
                       4087   4111#  4140  4165  4171  4178  4184  4193  4196  4220  4245  4251  4258
                       4264   4273  4281  4284  4308#  4337  4362  4368  4375  4381  4390  4393  4417
                       4442   4448  4455  4461  4470  4478  4481  4505#  4534  4559  4565  4572  4578
                       4587   4590  4614  4639  4645  4652  4658  4667  4675  4678  4702#  4731  4756
                       4762   4769  4775  4784  4787  4811  4836  4842  4849  4855  4864  4872  4875
                       4899#  4928  4953  4959  4966  4972  4981  4984  5008  5033  5039  5046  5052
                       5061   5069  5072  5097#  5125  5141  5144  5151  5154  5169  5172  5195  5211
                       5214   5221  5224  5239  5247  5250  5274#  5305  5333  5339  5346  5352  5361
```

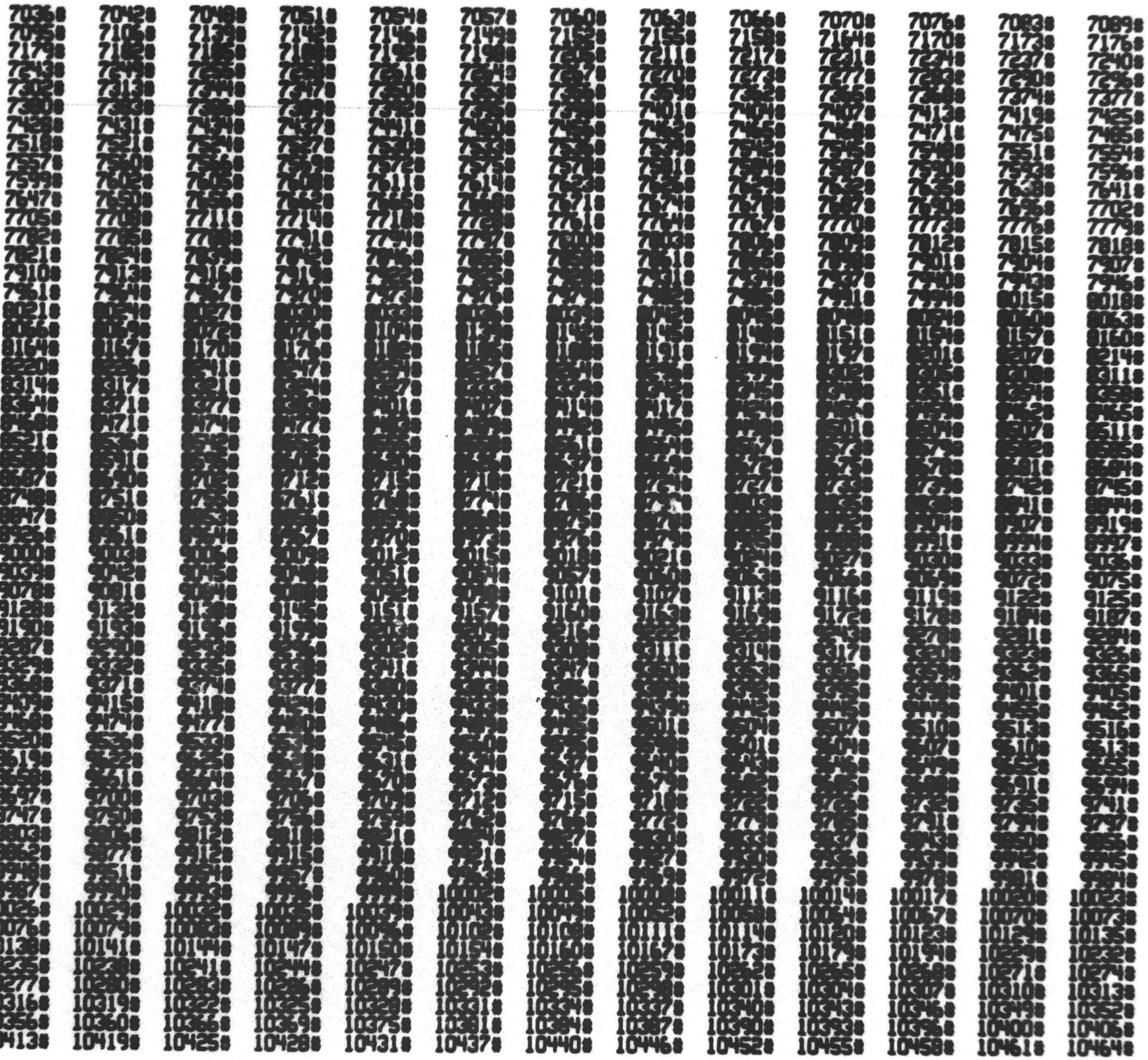
```
5364  5390  5418  5424  5431  5437  5446  5454  5457  5481#  5512  5540  5546
5553  5559  5568  5571  5597  5625  5631  5638  5644  5653  5661  5664  5688#
5719  5747  5753  5760  5766  5775  5778  5804  5832  5838  5845  5851  5860
5868  5871  5895#  5926  5954  5960  5967  5973  5982  5985  6011  6039  6045
6052  6058  6067  6075  6078  6102#  6133  6161  6167  6174  6180  6189  6192
6218  6246  6255  6259  6265  6274  6282  6285  6309#  6340  6368  6374  6381
6387  6396  6399  6425  6453  6459  6466  6472  6481  6489  6492  6516#  6547
6575  6581  6588  6594  6603  6608  6632  6660  6666  6673  6679  6688  6696
6699  6723#  6754  6782  6788  6795  6801  6810  6813  6839  6867  6873  6880
6886  6895  6903  6906  6930#  6961  6969  6975  7002  7008  7017  7020  7046
7074  7080  7087  7093  7102  7110  7113  7137#  7168  7196  7202  7209  7215
7224  7227  7253  7281  7287  7294  7300  7309  7317  7320  7343#  7411  7423
7445  7448  7460  7479  7482  7489  7492  7517#  7618  7645  7660  7682  7685
7700  7722  7725  7732  7735  7760#  7828  7834  7837  7850  7853  7869  7872
7897#  7956  7959  7989  8001  8013  8050  8060  8083  8089  8101  8108  8111
8137#  8180  8205  8211  8218  8224  8233  8251  8261  8268  8271  8294#  8337
8362  8369  8375  8381  8390  8393  8411  8421  8428  8431  8455#  8487  8515
8518  8525  8528  8546  8549  8577  8599  8602  8609  8612  8630  8633  8641
8644  8669#  8708  8709  8737  8753  8756  8772  8778  8796  8799  8807  8810
8834#  8872  8886  8895  8907  8923  8930  8933  8960#  9111  9136  9142
9149  9155  9182  9207  9213  9220  9226  9237  9240  9247  9250  9277#  9428
9453  9459  9466  9472  9479  9491  9530  9537  9543  9554  9557  9564  9567
9594#  9745  9770  9779  9783  9805  9816  9841  9847  9854  9860  9871  9874
9881  9884  9911#  10043  10087  10093  10100  10106  10133  10158  10164  10171  10177
10188  10191  10198  10201  10228#  10377  10404  10410  10417  10423  10450  10475  10481
10488  10494  10505  10508  10515  10518  10545#  10696  10721  10727  10734  10740  10767
10792  10798  10805  10811  10822  10825  10832  10835  10862#  11013  11038  11044  11051
11057  11084  11109  11115  11122  11128  11139  11142  11149  11152  11179#  11330  11355
11361  11368  11374  11401  11406  11432  11439  11445  11456  11459  11466  11469  11496#
11647  11672  11678  11685  11691  11718  11743  11749  11756  11762  11773  11776  11783
11786  11813#  11844  11865  11965  12002  12008  12035  12060  12066  12073  12079  12090
12093  12100  12103  12110  12261  12306  12312  12319  12352  12356  12377  12383  12390
12396  12407  12410  12417  12450  12457#  12508  12523  12530  12536  12642  12669  12694
12700  12707  12713  12724  12727  12734  12737  12764#  12915  12940  12946  12953  12959
12986  13011  13017  13024  13030  13041  13044  13051  13054  13081#  13232  13257  13263
13270  13276  13303  13308  13334  13341  13347  13358  13361  13368  13371  13398#  13549
13574  13580  13587  13593  13620  13645  13651  13658  13661  13675  13678  13685  13688
13713#  13733  13746  13749  13761  13768  13778  13781  13787  13799  13802  13816  13819
13826  13829  13856#  13884  13887  13894  14000  14007  14016  14019  14025  14041  14057
14067  14077  14087  14094  14107  14117  14129  14139  14149  14159  14169  14179  14169
14199  14215  14244  14247  14260  14273  14286  14293  14312  14325  14338  14365  14384
14387  14393  14396  14402  14405  14412  14415  14430  14452  14480#  14548  14560  14572
14578  14587  14593  14602  14617  14620  14635  14658  14661  14668  14671  14698#  14736
14742  14762  14765  14777  14780  14788  14808#  14849  14855  14858  14919  14925  14928
14983  14988  15038  15045  15051
```

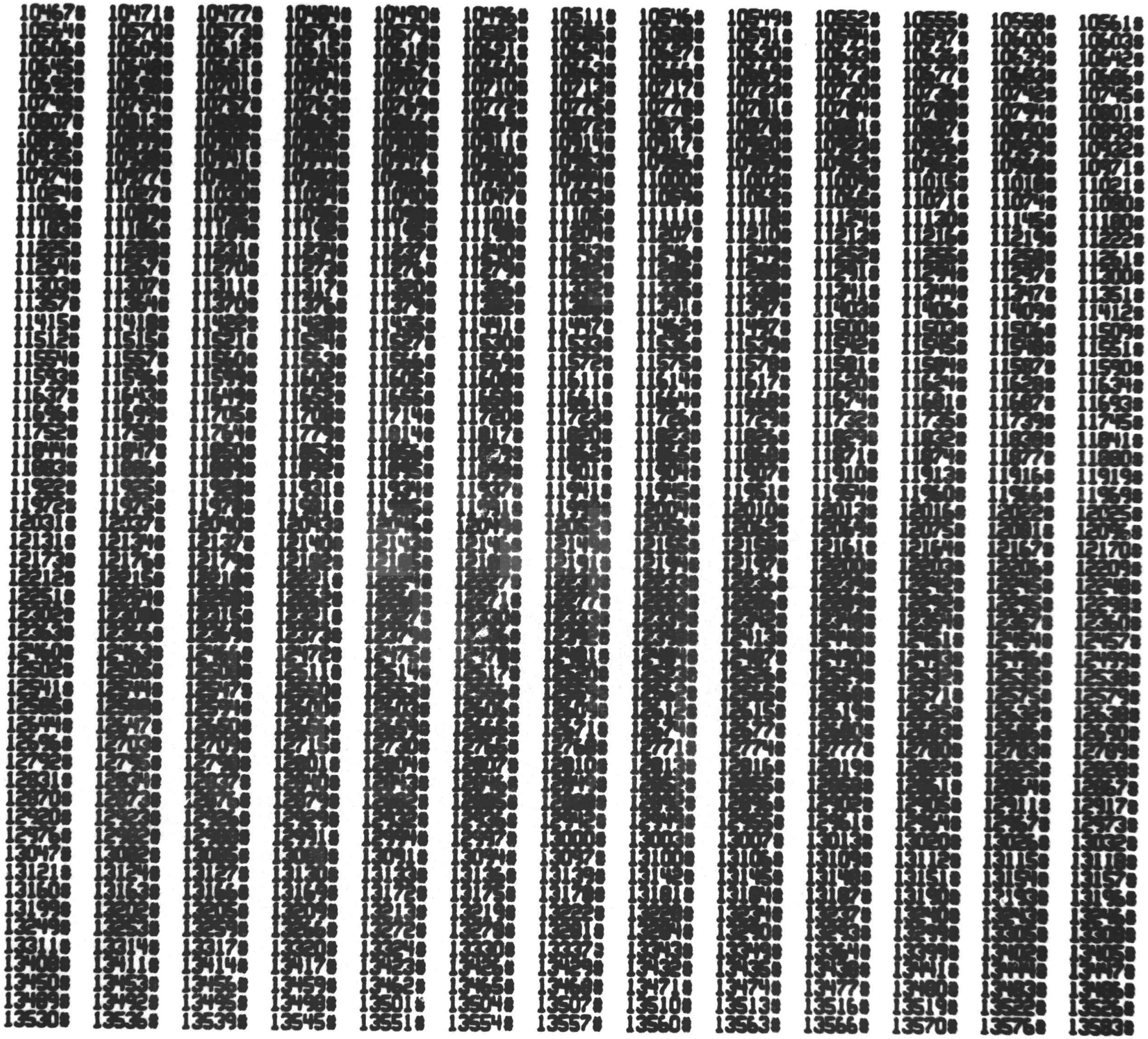
SSSDER= 000001

```
2283#  2286#  2290#  2294#  2297#  2300#  2306#  2312#  2315#  2318#  2321#  2324#  2327#
2331#  2337#  2344#  2350#  2358#  2370#  2374#  2377#  2380#  2386#  2392#  2395#  2398#
2401#  2404#  2407#  2411#  2417#  2424#  2430#  2436#  2447#  2480#  2483#  2487#  2491#
2494#  2497#  2503#  2509#  2512#  2515#  2518#  2521#  2524#  2528#  2534#  2541#  2547#
2553#  2567#  2571#  2574#  2577#  2583#  2589#  2592#  2596#  2598#  2601#  2604#  2608#
2614#  2621#  2627#  2633#  2644#  2677#  2680#  2684#  2688#  2691#  2694#  2700#  2706#
2709#  2712#  2715#  2718#  2721#  2725#  2731#  2738#  2744#  2760#  2764#  2768#  2771#
2774#  2780#  2786#  2789#  2792#  2795#  2798#  2801#  2805#  2811#  2818#  2824#  2830#
2841#  2874#  2877#  2881#  2885#  2888#  2891#  2897#  2903#  2906#  2909#  2912#  2915#
2918#  2922#  2928#  2935#  2941#  2947#  2961#  2965#  2968#  2971#  2977#  2983#  2986#
2989#  2992#  2995#  2998#  3002#  3008#  3015#  3021#  3027#  3038#  3072#  3075#  3079#
```

```
3083#  3086#  3089#  3092#  3095#  3098#  3104#  3110#  3113#  3116#  3119#  3122#  3125#
3129#  3135#  3142#  3148#  3154#  3160#  3172#  3175#  3178#  3181#  3184#  3187#  3193#
3199#  3202#  3205#  3208#  3211#  3214#  3218#  3224#  3231#  3237#  3243#  3254#  3269#
3292#  3296#  3300#  3303#  3306#  3309#  3312#  3315#  3318#  3321#  3324#  3330#  3336#
3339#  3342#  3345#  3348#  3351#  3355#  3361#  3368#  3374#  3380#  3394#  3398#  3401#
3404#  3407#  3410#  3413#  3416#  3419#  3422#  3428#  3434#  3437#  3440#  3443#  3446#
3449#  3453#  3459#  3466#  3472#  3478#  3489#  3522#  3525#  3529#  3533#  3536#  3539#
3545#  3551#  3554#  3557#  3560#  3563#  3566#  3570#  3576#  3583#  3589#  3595#  3609#
3613#  3616#  3619#  3625#  3631#  3634#  3637#  3640#  3643#  3646#  3650#  3655#  3663#
3669#  3675#  3686#  3719#  3722#  3726#  3730#  3733#  3736#  3742#  3748#  3751#  3754#
3757#  3760#  3763#  3767#  3773#  3780#  3786#  3792#  3800#  3810#  3813#  3816#  3822#
3828#  3831#  3834#  3837#  3840#  3843#  3847#  3853#  3860#  3866#  3872#  3883#  3916#
3919#  3923#  3927#  3930#  3933#  3937#  3941#  3945#  3948#  3951#  3954#  3957#  3960#  3964#
3970#  3977#  3983#  3987#  4003#  4007#  4010#  4013#  4019#  4025#  4028#  4031#  4034#
4037#  4040#  4044#  4050#  4057#  4063#  4069#  4080#  4113#  4116#  4120#  4124#  4127#
4130#  4136#  4142#  4145#  4148#  4151#  4154#  4157#  4161#  4167#  4174#  4180#  4186#
4200#  4204#  4207#  4210#  4216#  4222#  4225#  4228#  4231#  4234#  4237#  4241#  4247#
4254#  4260#  4266#  4277#  4310#  4313#  4317#  4321#  4324#  4327#  4333#  4339#  4342#
4345#  4349#  4351#  4354#  4358#  4364#  4371#  4377#  4383#  4397#  4401#  4404#  4407#
4413#  4419#  4422#  4425#  4428#  4431#  4434#  4438#  4444#  4451#  4457#  4463#  4474#
4507#  4510#  4514#  4518#  4521#  4524#  4530#  4536#  4539#  4542#  4545#  4548#  4551#
4555#  4561#  4564#  4574#  4580#  4594#  4598#  4601#  4604#  4610#  4616#  4619#  4622#
4625#  4628#  4631#  4635#  4641#  4648#  4654#  4660#  4671#  4704#  4707#  4711#  4715#
4718#  4721#  4727#  4733#  4736#  4739#  4742#  4745#  4748#  4752#  4758#  4765#  4771#
4777#  4791#  4795#  4798#  4801#  4807#  4813#  4816#  4819#  4822#  4825#  4828#  4832#
4838#  4845#  4851#  4857#  4860#  4901#  4904#  4908#  4912#  4915#  4918#  4924#  4930#
4933#  4936#  4939#  4942#  4945#  4949#  4955#  4962#  4968#  4974#  4988#  4992#  4995#
4998#  5004#  5010#  5012#  5016#  5019#  5022#  5025#  5029#  5035#  5042#  5048#  5054#
5065#  5099#  5102#  5106#  5109#  5112#  5115#  5121#  5127#  5130#  5133#  5137#  5147#
5156#  5165#  5169#  5173#  5182#  5186#  5191#  5197#  5200#  5203#  5207#  5217#  5226#
5236#  5243#  5247#  5250#  5253#  5256#  5259#  5262#  5265#  5268#  5271#  5310#  5313#
5316#  5319#  5322#  5326#  5329#  5332#  5335#  5338#  5342#  5345#  5348#  5351#  5377#
5380#  5386#  5389#  5392#  5395#  5398#  5401#  5404#  5407#  5410#  5414#  5420#  5427#  5433#
5439#  5450#  5453#  5456#  5459#  5462#  5465#  5468#  5471#  5474#  5478#  5481#  5484#
5523#  5526#  5529#  5532#  5535#  5538#  5541#  5544#  5547#  5550#  5553#  5556#  5559#  5562#
5587#  5590#  5593#  5596#  5599#  5602#  5605#  5608#  5611#  5614#  5617#  5621#  5624#  5627#  5630#  5634#  5637#  5640#
5646#  5657#  5660#  5663#  5666#  5669#  5672#  5675#  5678#  5681#  5684#  5687#  5690#  5693#  5696#  5699#  5702#  5705#  5708#  5711#  5714#  5715#  5718#  5721#  5724#  5727#
5730#  5733#  5736#  5739#  5742#  5745#  5748#  5751#  5754#  5757#  5760#  5763#  5766#  5769#  5772#  5775#  5778#  5781#  5784#  5787#  5790#  5793#  5796#  5799#  5802#  5805#  5808#  5811#  5814#  5817#  5820#  5823#  5826#  5829#  5832#  5835#  5838#  5841#  5844#  5847#
5794#  5800#  5806#  5809#  5812#  5815#  5818#  5821#  5824#  5827#  5830#  5833#  5836#  5839#  5842#  5845#  5848#  5851#  5854#  5857#  5860#  5863#  5866#  5869#  5872#  5875#  5878#  5881#  5884#  5887#  5890#  5893#  5896#  5899#  5902#  5905#  5908#  5911#  5914#  5917#  5920#  5923#  5926#  5929#  5932#  5935#  5938#  5941#  5944#  5947#
5853#  5864#  5867#  5870#  5873#  5876#  5879#  5882#  5885#  5888#  5891#  5894#  5897#  5900#  5903#  5906#  5909#  5912#  5915#  5918#  5921#  5924#  5927#  5930#  5933#  5936#  5939#  5942#  5945#  5948#  5951#  5954#  5957#  5960#  5963#  5966#  5969#  5972#  5975#  5978#  5981#  5984#  5987#  5990#  5993#  5996#  5999#  6002#  6005#  6008#  6011#  6014#  6017#  6020#  6023#  6026#  6029#  6032#  6035#  6038#  6041#  6044#  6047#  6050#  6053#  6056#  6059#  6062#  6065#  6068#  6071#  6074#  6077#  6080#  6083#  6086#  6089#  6092#  6095#  6098#  6101#  6104#  6107#  6110#  6113#  6116#  6119#  6122#  6125#  6128#  6131#  6134#  6137#  6140#  6143#  6146#  6149#  6152#  6155#  6158#  6161#  6164#  6167#  6170#  6173#  6176#  6179#  6182#  6185#  6188#  6191#  6194#  6197#  6200#  6203#  6206#  6209#  6212#  6215#  6218#  6221#  6224#  6227#  6230#  6233#  6236#  6239#  6242#  6245#  6248#  6251#  6254#  6257#  6260#  6263#  6266#  6269#  6272#  6275#  6278#  6281#  6284#  6287#  6290#  6293#  6296#  6299#  6302#  6305#  6308#  6311#  6314#  6317#  6320#  6323#  6326#  6329#  6332#  6335#  6338#  6341#  6344#  6347#  6350#  6353#  6356#  6359#  6362#  6365#  6368#  6371#  6374#  6377#  6380#  6383#  6386#  6389#  6392#  6395#  6398#  6401#  6404#  6407#  6410#  6413#  6416#  6419#  6422#  6425#  6428#  6431#  6434#  6437#  6440#  6443#  6446#  6449#  6452#  6455#  6458#  6461#  6464#  6467#  6470#  6473#  6476#  6479#  6482#  6485#  6488#  6491#  6494#  6497#  6500#  6503#  6506#  6509#  6512#  6515#  6518#  6521#  6524#  6527#  6530#  6533#  6536#  6539#  6542#  6545#  6548#  6551#  6554#  6557#  6560#  6563#  6566#  6569#  6572#  6575#  6578#  6581#  6584#  6587#  6590#  6593#  6596#  6599#  6602#  6605#  6608#  6611#  6614#  6617#  6620#  6623#  6626#  6629#  6632#  6635#  6638#  6641#  6644#  6647#  6650#  6653#  6656#  6659#  6662#  6665#  6668#  6671#  6674#  6677#  6680#  6683#  6686#  6689#  6692#  6695#  6698#  6701#  6704#  6707#  6710#  6713#  6716#  6719#  6722#  6725#  6728#  6732#  6735#  6738#  6741#  6744#  6747#  6750#  6753#  6756#  6759#  6762#  6765#  6768#  6771#  6774#  6778#  6784#  6791#  6797#  6803#  6817#  6820#  6823#  6826#  6829#  6835#  6841#  6844#  6847#  6850#  6853#  6856#  6859#  6863#  6869#  6876#  6882#  6888#  6899#  6932#  6935#  6939#  6942#  6945#  6948#  6951#  6957#  6963#  6966#  6969#  6972#  6975#  6978#  6981#  6985#  6991#  6998#  7004#  7010#  7024#  7027#  7030#  7033#
```

CROSS REFERENCE TABLE -- USER SYMBOLS





```
                13589#  13595#  13598#  13601#  13607#  13610#  13616#  13622#  13625#  13628#  13631#  13634#  13637#
                13641#  13647#  13654#  13660#  13666#  13691#  13714#  13717#  13720#  13723#  13726#  13729#  13735#
                13738#  13742#  13751#  13754#  13757#  13767#  13770#  13774#  13783#  13787#  13790#  13793#  13812#
                13822#  13827#  13840#  13843#  13850#  13870#  13873#  13877#  13880#  13883#  13887#  13890#  13892#
                13895#  13900#  13903#  13906#  13909#  13912#  13915#  13918#  13921#  13927#  13930#  13933#  13936#
                13939#  13942#  13945#  13948#  13951#  13954#  13957#  13962#  13971#  13974#  13980#  13983#  13986#
                14009#  14021#  14027#  14031#  14034#  14046#  14049#  14053#  14056#  14063#  14067#  14073#  14079#
                14083#  14089#  14093#  14097#  14103#  14109#  14113#  14121#  14125#  14131#  14135#  14141#  14145#
                14151#  14155#  14161#  14165#  14171#  14175#  14181#  14185#  14191#  14195#  14201#  14207#  14211#
                14217#  14220#  14230#  14236#  14240#  14246#  14249#  14253#  14259#  14262#  14268#  14272#  14275#
                14279#  14282#  14286#  14289#  14292#  14295#  14301#  14304#  14308#  14314#  14317#  14321#
                14327#  14330#  14334#  14337#  14340#  14343#  14346#  14349#  14352#  14361#  14367#  14370#  14373#  14376#
                14380#  14408#  14417#  14420#  14423#  14426#  14429#  14432#  14435#  14441#  14444#  14448#  14481#  14484#
                14487#  14490#  14493#  14496#  14499#  14502#  14505#  14508#  14511#  14514#  14517#  14520#  14523#
                14529#  14532#  14535#  14538#  14541#  14544#  14547#  14550#  14553#  14556#  14562#  14565#  14568#
                14574#  14580#  14583#  14586#  14589#  14592#  14595#  14598#  14601#  14604#  14607#  14612#  14623#  14626#  14632#  14638#
                14641#  14644#  14647#  14650#  14653#  14656#  14659#  14662#  14665#  14668#  14671#  14674#  14677#
                14720#  14723#  14726#  14729#  14732#  14735#  14738#  14741#  14747#  14750#  14753#  14756#  14759#  14762#  14765#  14768#  14771#  14774#  14777#  14780#  14783#  14786#  14789#  14792#  14795#  14798#  14801#  14804#  14807#  14810#  14813#  14816#  14819#  14822#  14825#  14828#  14831#  14834#  14837#  14840#  14843#  14846#  14849#  14852#  14855#  14858#  14861#  14864#  14867#  14870#  14873#  14876#  14879#
                14882#  14885#  14888#  14891#  14894#  14897#  14900#  14903#  14906#  14909#  14912#  14915#  14921#  14930#
                14933#  14936#  14939#  14942#  14945#  14948#  14951#  14954#  14957#  14960#  14963#  14966#  14969#  14972#  14975#  14978#  14981#  14984#  14987#  14990#  14993#  14996#  14999#  15002#  15005#  15008#  15011#  15014#  15017#  15020#  15023#  15026#  15029#  15032#  15035#  15038#  15041#
                15044#  15047#
SSSSER= 000001  2283#  2285  2286#  2288  2290#  2292  2294#  2296  2297#  2299  2300#  2302  2312#
                2314  2315#  2317  2318#  2320  2321#  2323  2324#  2326  2327#  2329  2331#  2333
                2337#  2339  2344#  2346  2350#  2352  2356#  2357  2358#  2370#  2372  2374#  2376  2377#
                2379  2380#  2382  2382#  2384  2385#  2387  2388#  2400  2401#  2403  2404#  2406
                2407#  2409  2411#  2413  2417#  2419  2424#  2426  2430#  2432  2436#  2438  2447#
                2449  2480#  2482  2483#  2485  2487#  2489  2491#  2493  2494#  2523  2496  2497#  2499
                2509#  2511  2512#  2514  2515#  2517  2518#  2520  2521#  2523  2524#  2526  2571#  2528#
                2530  2534#  2536  2541#  2543  2547#  2549  2553#  2555  2557  2559  2569  2591#  2573
                2574#  2576  2577#  2579  2589#  2591  2592#  2593#  2594  2597  2598#  2600  2720  2601#
                2603  2604#  2606  2608#  2610  2614#  2616  2621#  2623  2627  2629  2633#  2764#  2635
                2644#  2646  2677#  2673  2680#  2682  2684#  2686  2688#  2690  2691#  2693  2694#
                2696  2706#  2708  2709#  2711  2712#  2714  2715#  2717  2718#  2720  2721#  2723
                2725#  2727  2731#  2733  2738#  2740  2744#  2746  2750#  2752  2764#  2766  2768#
                2770  2771#  2773  2774#  2776  2786#  2788  2789#  2791  2792#  2794  2795#  2797
                2798#  2800  2801#  2803  2805#  2807  2811#  2813  2818#  2820  2824#  2826  2830#
                2832  2841#  2843  2874#  2876  2877#  2879  2881#  2883  2885#  2887  2888#  2890
                2891#  2893  2903#  2905  2906#  2908  2909#  2911  2912#  2914  2915#  2917  2918#
                2920  2922#  2924  2926#  2930  2931#  2937  2941#  2943  2947#  2949  2961#  2963
                2965#  2967  2969#  2970  2971#  2973#  2983#  2985  2986#  2988  2989#  2991  2992#
                2994  2996#  2998  3000  3072#  3074  3075#  3077  3010  3079#  3017  3021#  3023
                3027#  3029  3038#  3040  3045#  3098#  3100  3110#  3015#  3081  3083#  3085  3086#
                3088  3089#  3091  3095#  3124  3127  3129  3131  3112  3135#  3115  3116#  3118
                3119#  3121  3122#  3126#  3170  3172#  3174  3175#  3137  3177  3142#  3144  3148#
                3150  3154#  3156  3160#  3201  3204  3205#  3207  3178#  3208#  3180  3184#  3186
                3187#  3189  3199#  3224#  3231#  3233  3237#  3210  3239  3211#  3213  3214#
                3216  3218#  3220  3222#  3264  3266#  3300#  3302  3243#  3303#  3245  3254#  3256
                3289#  3291  3292#  3294  3296#  3298#  3321#  3323  3305  3338  3306#  3308  3312#
                3314  3315#  3317  3318#  3320  3351#  3353  3355#  3336#  3357  3339#  3341  3342#  3344
                3345#  3347  3348#  3350  3394#  3396  3398#  3400  3363  3401#  3404#  3368#  3370  3374#
                3376  3380#  3382  3394#  3396  3398#  3400  3401#  3403  3404#  3406  3410#  3412
                3413#  3415  3416#  3418  3419#  3421  3434#  3436  3437#  3439  3440#  3442  3443#
                3445  3446#  3448  3449#  3451  3453#  3455  3459#  3461  3466#  3468  3472#  3474
                3478#  3480  3489#  3491  3522#  3524  3525#  3527  3529#  3531  3533#  3535  3536#
```

```
3538    3539#   3541    3551#   3553    3554#   3556    3557#   3559    3560#   3562    3563#   3565
3566#   3568    3570#   3572    3576#   3578    3583#   3585    3589#   3591    3595#   3597    3609#
3611    3613#   3615    3616#   3618    3619#   3621    3631#   3633    3634#   3636    3637#   3639
3640#   3642    3643#   3645    3646#   3648    3650#   3652    3656#   3658    3663#   3665    3669#
3671    3675#   3677    3686#   3688    3719#   3721    3722#   3724    3726#   3728    3730#   3732
3733#   3735    3736#   3738    3748#   3750    3751#   3753    3754#   3756    3757#   3759    3760#
3762    3763#   3766    3767#   3769    3773#   3775    3780#   3782    3786#   3788    3792#   3794
3806#   3808    3810#   3812    3813#   3815    3816#   3818    3828#   3830    3831#   3833    3834#
3836    3837#   3839    3840#   3842    3843#   3845    3847#   3849    3853#   3855    3860#   3862
3864#   3868    3872#   3874    3883#   3885    3916#   3918    3919#   3921    3923#   3925    3927#
3929    3930#   3932    3933#   3935    3945#   3947    3948#   3950    3951#   3953    3954#   3956
3957#   3959    3960#   3965    3966    3970#   3972    3977#   3979    3983#   3985    3988#   3989#
3991    4003#   4005    4007#   4009    4010#   4012    4013#   4015    4025#   4027    4028#   4030
4031#   4033    4034#   4036    4037#   4039    4040#   4042    4044#   4046    4050#   4052    4057#
4059    4063#   4065    4069#   4071    4080#   4082    4113#   4115    4116#   4118    4120#   4122
4124#   4126    4127#   4129    4130#   4132    4142#   4144    4145#   4147    4148#   4150    4151#
4153    4154#   4156    4157#   4159    4161#   4163    4167#   4169    4174#   4176    4180#   4182
4186#   4188    4200#   4202    4204#   4206    4207#   4209    4210#   4212    4222#   4224    4225#
4227    4228#   4230    4231#   4233    4234#   4236    4237#   4239    4241#   4243    4247#   4249
4254#   4256    4260#   4262    4266#   4268    4277#   4279    4310#   4312    4313#   4315    4317#
4319    4321#   4323    4324#   4326    4327#   4329    4339#   4341    4342#   4344    4345#   4347
4348#   4350    4351#   4353    4354#   4356    4358#   4360    4364#   4366    4371#   4373    4377#
4379    4383#   4385    4397#   4399    4401#   4403    4404#   4406    4407#   4409    4419#   4421
4422#   4424    4425#   4427    4428#   4430    4431#   4433    4434#   4436    4438#   4440    4444#
4446    4451#   4453    4457#   4459    4463#   4465    4474#   4476    4507#   4509    4510#   4512
4514#   4516    4518#   4520    4521#   4523    4524#   4526    4536#   4538    4539#   4541    4542#
4544    4545#   4547    4548#   4550    4551#   4553    4555#   4557    4561#   4563    4568#   4570
4574#   4576    4580#   4582    4594#   4596    4598#   4600    4601#   4603    4604#   4606    4616#
4618    4619#   4621    4622#   4624    4625#   4627    4628#   4630    4631#   4633    4635#   4637
4641#   4643    4648#   4650    4654#   4656    4660#   4662    4671#   4673    4704#   4706    4707#
4709    4711#   4713    4715#   4717    4718#   4720    4721#   4723    4733#   4735    4736#   4738
4739#   4741    4742#   4744    4745#   4747    4748#   4750    4752#   4754    4758#   4760    4765#
4767    4771#   4773    4777#   4779    4791#   4793    4795#   4797    4798#   4800    4801#   4803
4813#   4815    4816#   4818    4819#   4821    4822#   4824    4825#   4827    4828#   4830    4832#
4834    4838#   4840    4845#   4847    4851#   4853    4857#   4859    4868#   4870    4901#   4903
4904#   4906    4908#   4910    4912#   4914    4915#   4917    4918#   4920    4930#   4932    4933#
4935    4936#   4938    4939#   4941    4942#   4944    4945#   4947    4949#   4951    4955#   4957
4962#   4964    4968#   4970    4974#   4976    4988#   4990    4992#   4994    4996#   4997    4998#
5000    5010#   5012    5013#   5015    5016#   5018    5019#   5021    5022#   5024    5025#   5027
5029#   5031    5035#   5037    5042#   5044    5048#   5050    5054#   5056    5065#   5067    5099#
5101    5102#   5104    5106#   5108    5109#   5111    5112#   5114    5115#   5117    5127#   5129
5130#   5132    5133#   5135    5137#   5139    5147#   5149    5156#   5158    5165#   5167    5176#
5178    5179#   5181    5182#   5184    5185#   5187    5197#   5199    5200#   5202    5203#   5205
5207#   5209    5217#   5219    5226#   5228    5235#   5237    5243#   5245    5276#   5278    5279#
5281    5283#   5285    5286#   5288    5289#   5291    5292#   5294    5295#   5297    5307#   5309
5310#   5312    5313#   5315    5316#   5318    5319#   5321    5322#   5324    5325#   5327    5329#
5331    5335#   5337    5342#   5344    5348#   5350    5354#   5356    5360#   5370    5371#   5373
5374#   5376    5377#   5379    5380#   5382    5392#   5394    5395#   5397    5398#   5400    5401#
5403    5404#   5406    5407#   5409    5410#   5412    5414#   5416    5420#   5422    5427#   5429
5433#   5435    5439#   5441    5450#   5452    5483#   5485    5486#   5488    5490#   5492    5493#
5495    5496#   5498    5499#   5501    5502#   5504    5514#   5516    5517#   5519    5520#   5522
5523#   5525#   5526#   5528    5529#   5531    5532#   5534    5536#   5538    5542#   5544    5549#
5551    5555#   5557    5561#   5563    5575#   5577    5578#   5580    5581#   5583    5584#   5586
5587#   5589    5599#   5601    5602#   5604    5605#   5607    5608#   5610    5611#   5613    5614#
5616    5617#   5619    5621#   5623    5627#   5629    5634#   5636    5640#   5642    5646#   5648
5657#   5659    5690#   5692    5693#   5695    5697#   5699    5700#   5702    5703#   5705    5706#
```

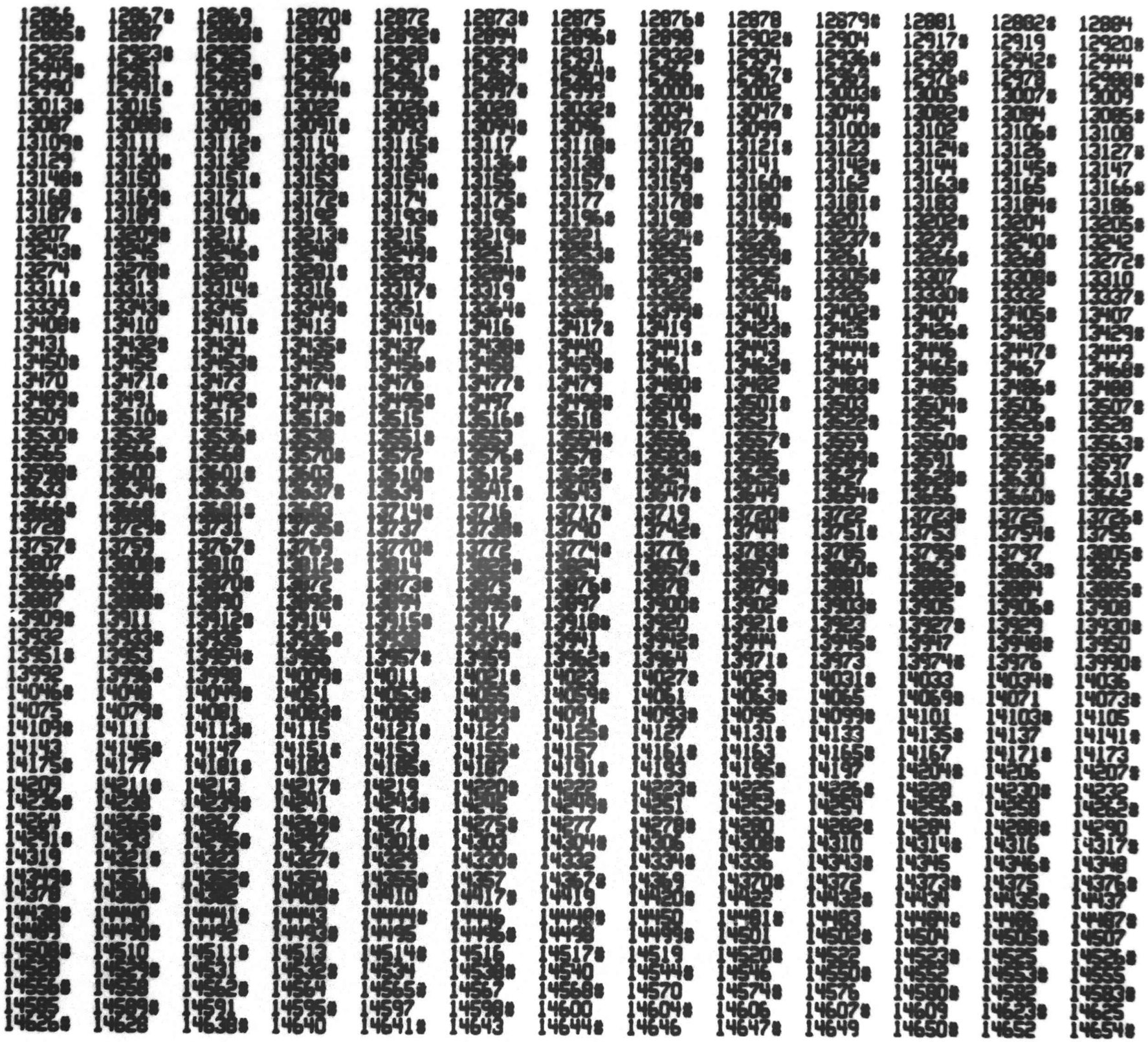
F10


```
5708    5709#   5711    5721#   5723    5724#   5726    5727#   5729    5730#   5732    5733#   5735
5736#   5738    5739#   5741    5743#   5745    5749#   5751    5756#   5758    5762#   5764    5768#
5770    5782#   5784    5785#   5787    5788#   5790    5791#   5793    5794#   5796    5806#   5808
5809#   5811    5812#   5814    5815#   5817    5818#   5820    5821#   5823    5824#   5826    5828#
5830    5834#   5836    5841#   5843    5847#   5849    5853#   5855    5864#   5866    5897#   5899
5900#   5902    5904#   5906    5907#   5909    5910#   5912    5913#   5915    5916#   5918    5928#
5940    5941#   5943    5944#   5946    5957#   5959    5937#   5939    5940#   5942    5943#   5945    5946#   5948
5950#   5952    5956#   5958    5963#   5965    5969#   5971    5975#   5977    5989#   5991    5992#
5994    5998#   5997    5998#   6000    6001#   6003    6013#   6015    6016#   6018    6019#   6021
6022#   6024    6025#   6027    6028#   6030    6031#   6033    6035#   6037    6041#   6043    6048#
6050    6054#   6056    6060#   6062    6071#   6073    6104#   6106    6107#   6109    6111#   6113
6114#   6116    6117#   6119    6120#   6122    6123#   6125    6136#   6137    6138#   6140    6141#
6143    6144#   6146    6147#   6149    6150#   6152    6153#   6155    6157#   6159    6163#   6165
6170#   6172    6176#   6178    6182#   6184    6196#   6198    6199#   6201    6202#   6204    6205#
6207    6208#   6210    6220#   6222    6223#   6225    6226#   6228    6229#   6231    6232#   6234
6235#   6237    6238#   6240    6242#   6244    6248#   6250    6265#   6267    6261#   6263    6267#
6269    6278#   6280    6311#   6313    6314#   6316    6318#   6320    6321#   6323    6324#   6326
6327#   6329    6330#   6332    6342#   6344    6345#   6347    6348#   6350    6351#   6353    6354#
6356    6357#   6359    6360#   6362    6364#   6366    6370#   6372    6377#   6379    6383#   6385
6389#   6391    6403#   6405    6406#   6408    6409#   6411    6412#   6414    6415#   6417    6427#
6429    6430#   6432    6433#   6435    6436#   6438    6439#   6441    6442#   6444    6445#   6447
6449#   6451    6456#   6457    6462#   6464    6468#   6470    6474#   6476    6485#   6487    6518#
6520    6521#   6523    6525#   6527    6528#   6530    6531#   6533    6534#   6536    6537#   6539
6549#   6551    6556#   6561    6565#   6567    6568#   6560    6561#   6563    6564#   6566    6567#
6569    6571#   6573    6577#   6579    6584#   6586    6530#   6592    6596#   6598    6610#   6612
6613#   6615    6616#   6618    6619#   6621    6622#   6624    6634#   6636    6637#   6639    6640#
6642    6643#   6645    6646#   6648    6649#   6651    6652#   6654    6656#   6658    6662#   6664
6669#   6671    6675#   6677    6681#   6683    6684#   6691    6725#   6727    6728#   6730    6732#
6734    6735#   6737    6738#   6740    6741#   6743    6744#   6746    6756#   6758    6759#   6761
6762#   6764    6765#   6767    6768#   6770    6771#   6773    6774#   6776    6778#   6780    6784#
6786    6791#   6793    6797#   6799    6803#   6805    6817#   6819    6820#   6822    6823#   6825
6826#   6828    6829#   6831    6841#   6843    6844#   6846    6847#   6849    6850#   6852    6853#
6855    6856#   6858    6859#   6861    6863#   6865    6869#   6871    6876#   6878    6882#   6884
6888#   6890    6894#   6901    6932#   6934    6935#   6937    6939#   6941    6942#   6944    6945#
6947    6948#   6950    6951#   6953    6963#   6965    6966#   6968    6969#   6971    6972#   6974
6975#   6977    6978#   6980    6981#   6983    6985#   6987    6991#   6993    6998#   7000    7004#
7006    7010#   7012    7024#   7026    7027#   7029    7030#   7032    7033#   7035    7036#   7038
7048#   7050    7051#   7053    7054#   7056    7057#   7059    7060#   7062    7063#   7065    7066#
7068    7070#   7072    7076#   7078    7083#   7085    7089#   7091    7095#   7097    7105#   7108
7139#   7141    7142#   7144    7146#   7148    7149#   7151    7152#   7154    7155#   7157    7158#
7160    7170#   7172    7173#   7175    7176#   7178    7179#   7181    7182#   7184    7185#   7187
7188#   7190    7192#   7194    7198#   7200    7205#   7207    7211#   7213    7217#   7219    7231#
7233    7234#   7236    7237#   7239    7240#   7242    7243#   7245    7255#   7257    7258#   7260
7261#   7263    7264#   7266    7267#   7269    7270#   7272    7273#   7275    7277#   7279    7283#
7285    7290#   7292    7296#   7298    7302#   7304    7313#   7315    7344#   7346    7347#   7349
7350#   7352    7353#   7355    7356#   7359    7359#   7361    7362#   7364    7365#   7367    7368#
7370    7374#   7376    7377#   7379    7380#   7382    7383#   7385    7386#   7388    7389#   7391
7392#   7394    7395#   7397    7398#   7400    7401#   7403    7404#   7406    7407#   7409    7413#
7415    7425#   7427    7428#   7430    7431#   7433    7434#   7436    7437#   7439    7441#   7443
7450#   7452    7462#   7464    7465#   7467    7468#   7470    7471#   7473    7475#   7477    7485#
7487    7518#   7520    7521#   7523    7524#   7526    7527#   7529    7530#   7532    7533#   7535
7536#   7538    7539#   7541    7542#   7544    7545#   7547    7548#   7550    7551#   7553    7557#
7559    7560#   7562    7566#   7568    7569#   7571    7572#   7574    7575#   7577    7578#   7580
7581#   7583    7584#   7586    7587#   7589    7590#   7592    7593#   7595    7596#   7598    7599#
7601    7602#   7604    7605#   7607    7608#   7610    7614#   7616    7623#   7625    7626#   7628
7629#   7631    7632#   7634    7635#   7637    7638#   7640    7641#   7643    7647#   7649    7650#
```

```
7652    7662#   7664    7665#   7667    7668#   7670    7671#   7673    7674#   7676    7678#   7680
7687#   7689    7690#   7692    7702#   7704    7705#   7707    7708#   7710    7711#   7713    7714#
7716    7718#   7720    7728#   7730    7761#   7763    7764#   7766    7767#   7769    7770#   7772
7773#   7775    7776#   7778    7779#   7781    7782#   7784    7785#   7787    7788#   7790    7791#
7793    7794#   7796    7797#   7799    7800#   7802    7803#   7805    7806#   7808    7809#   7811
7812#   7814    7818#   7820    7821#   7823    7824#   7826    7839#   7841    7842#   7844    7846#
7848    7855#   7857    7858#   7860    7865#   7867    7898#   7900    7901#   7903    7904#   7906
7907#   7909    7910#   7912    7913#   7915    7916#   7918    7919#   7921    7922#   7924    7925#
7927    7928#   7930    7931#   7933    7934#   7936    7937#   7939    7940#   7942    7943#   7945
7946#   7948    7961#   7963    7964#   7966    7967#   7969    7970#   7972    7973#   7975    7976#
7978    7979#   7981    7982#   7984    7985#   7987    7991#   7993    7994#   7996    8015#   8017
8018#   8020    8021#   8023    8024#   8026    8027#   8029    8030#   8032    8033#   8035    8036#
8038    8039#   8041    8042#   8044    8045#   8047    8048#   8050    8060#   8062    8063#   8065
8066#   8068    8069#   8071    8072#   8074    8076#   8078    8104#   8106    8139#   8141    8142#
8144    8145#   8147    8148#   8150    8151#   8153    8154#   8156    8157#   8159    8160#   8162
8164#   8166    8167#   8169    8170#   8172    8182#   8184    8185#   8187    8188#   8190    8191#
8193    8194#   8196    8197#   8199    8201#   8203    8207#   8209    8214#   8216    8220#   8222
8226#   8228    8241#   8243    8254#   8256    8257#   8259    8264#   8266    8296#   8298    8299#
8301    8302#   8304    8305#   8307    8308    8310    8311#   8313    8314#   8316    8317#   8319
8321#   8323    8324#   8326    8327#   8328    8339#   8341    8342#   8344    8345#   8347    8348#
8350    8351#   8353    8354#   8356    8358    8360    8364#   8366    8371#   8373    8377#   8379
8383#   8385    8401#   8403    8414#   8416    8417#   8419    8424#   8426    8456#   8458    8459#
8461    8462#   8464    8465#   8467    8468    8470    8471#   8473    8474#   8476    8477#   8479
8489#   8491    8492#   8494    8495#   8497    8498#   8500    8501#   8503    8504#   8506    8507#
8509    8511#   8513    8521#   8523    8556#   8558    8562#   8564    8555#   8557    8558#   8560
8561#   8563    8564#   8566    8567#   8569    8574#   8581    8582#   8584    8585#   8587    8588#
8590    8591#   8593    8595#   8597    8605#   8607    8620#   8622    8637#   8639    8669#   8671
8672#   8674    8675#   8677    8678#   8680    8681#   8683    8684#   8686    8687#   8689    8690#
8692    8712#   8714    8715#   8717    8718#   8720    8721#   8723    8724#   8726    8727#   8729
8739#   8741    8742#   8744    8745#   8747    8748#   8750    8751#   8753    8755#   8757    8761#
8763    8768#   8770    8774#   8776    8786#   8788    8803#   8805    8835#   8837    8838#   8840
8841#   8843    8844#   8846    8847#   8849    8850    8852    8853#   8855    8856#   8858    8859#
8861    8868#   8870    8875#   8877    8878#   8880    8882#   8884    8892#   8894    8904#   8906
8907#   8909    8919#   8921    8926#   8928    8951#   8953    8954#   8956    8967#   8969    8991#   8970#
8972    8973#   8975    8976#   8978    8979#   8981    8985#   8987    9008#   8990    8993    8993
8994#   8996    8997#   8999    9000#   9002    9003#   9005    9006#   9008    9009#   9011    9012#
9014    9015#   9017    9018#   9020    9021#   9023    9024#   9026    9027#   9029    9030#   9032    9032
9033#   9035    9036#   9038    9039#   9041    9042#   9044    9045#   9047    9047    9048#   9050    9051#
9053    9054#   9056    9057#   9059    9060#   9062    9063#   9065    9066#   9068    9069#   9071    9071
9072#   9074    9075#   9077    9078#   9080    9081#   9083    9084#   9086    9088#   9090    9092#   9092#
9094    9098#   9100    9101#   9103    9113#   9115    9116#   9118    9119#   9121    9122#   9124    9124
9125#   9127    9128#   9130    9132#   9134    9138#   9140    9145#   9147    9151#   9153    9157#   9157#
9159    9160#   9162    9163#   9165    9169#   9171    9172#   9174    9184#   9186    9187#   9189    9189
9190#   9192    9193#   9195    9196#   9198    9199#   9201    9203#   9205    9209#   9211    9216#   9216#
9218    9222#   9224    9228#   9230    9243#   9245    9278#   9280    9281#   9283    9284#   9286    9286
9287#   9289    9290#   9292    9293#   9295    9296#   9298    9302#   9304    9305#   9307    9308#   9308#
9310    9311#   9313    9314#   9316    9317#   9319    9320#   9322    9323#   9325    9326#   9328    9328
9329#   9331    9332#   9334    9335#   9337    9338#   9340    9341#   9343    9344#   9346    9347#   9347#
9349    9350#   9352    9353#   9355    9356#   9358    9359#   9361    9362#   9364    9365#   9367    9367
9368#   9370    9371#   9373    9374#   9376    9377#   9379    9380#   9382    9383#   9385    9386#   9386#
9388    9389#   9391    9392#   9394    9395#   9397    9398#   9400    9401#   9403    9405#   9407    9407
9409#   9411    9415#   9417    9430#   9432    9433#   9435    9436#   9438    9439#   9441    9442#   9442#
9444    9445#   9447    9449#   9451    9455#   9457    9462#   9464    9468#   9470    9474#   9476    9476
9477#   9479    9480#   9482    9489#   9491    9501#   9503    9504#   9506    9507#   9509    9510#   9510#
9512    9513#   9515    9516#   9518    9520#   9522    9526#   9528    9533#   9535    9539#   9541    9541
9545#   9547    9560#   9562    9595#   9597    9598#   9600    9601#   9603    9604#   9606    9607#   9607#
```

```
9609    9610#   9612    9613#   9615    9619#   9621    9622#   9624    9625#   9627    9628#   9630
9631#   9633    9634#   9636    9637#   9639    9640#   9642    9643#   9645    9646#   9648    9649#
9651    9652#   9654    9656#   9657    9658#   9660    9661#   9663    9664#   9666    9667#   9669
9670#   9672    9673#   9675    9676#   9678    9679#   9681    9682#   9684    9685#   9687    9688#
9690    9691#   9693    9694#   9696    9697#   9699    9700#   9702    9703#   9705    9706#   9708
9709#   9711    9712#   9714    9715#   9717    9718#   9720    9722#   9724    9726#   9728    9732#
9734    9747#   9749    9750#   9752    9753#   9755    9756#   9758    9759#   9761    9762#   9764
9766#   9768    9772#   9774    9776#   9781    9785#   9787    9791#   9793    9794#   9796    9797#
9799    9806#   9809    9818#   9820    9821#   9823    9824#   9826    9827#   9829    9830#   9832
9833#   9835    9837#   9839    9843#   9845    9850#   9852    9856#   9858    9862#   9864    9877#
9879    9912#   9914    9915#   9917    9918#   9920    9921#   9923    9924#   9926    9927#   9929
9930#   9932    9936#   9938    9939#   9941    9942#   9944    9945#   9947    9948#   9950    9951#
9953    9954    9956    9957#   9959    9960    9961#   9963#   9965    9966#   9968    9969#   9971
9972#   9974    9975#   9977    9978#   9980    9981#   9983    9984#   9986    9987#   9989    9990#
9992    9993#   9995    9996#   9998    9999#   10001   10002#  10004   10005#  10007   10008#  10010
10011#  10013   10014#  10016   10017#  10019   10020#  10022   10023#  10025   10026#  10028   10029#
10031   10032#  10034   10035#  10037   10039#  10041   10043#  10045   10049#  10051   10064#  10066
10067#  10069   10070#  10072   10073#  10075   10076#  10078   10079#  10081   10083#  10085   10089#
10091   10096#  10098   10102#  10104   10108#  10110   10111#  10113   10114#  10116   10123#  10125
10135#  10137   10138#  10140   10141#  10143   10144#  10146   10147#  10149   10150#  10152   10154#
10156   10160#  10162   10167#  10169   10173#  10175   10179#  10181   10194#  10196   10229#  10231
10232#  10234   10235#  10237   10238#  10240   10241#  10243   10244#  10246   10247#  10249   10253#
10255   10256#  10258   10259#  10261   10262#  10264   10265#  10267   10268#  10270   10271#  10273
10274#  10276   10277#  10279   10280#  10282   10283#  10285   10286#  10288   10289#  10291   10292#
10294   10295#  10297   10298#  10300   10301#  10303   10304#  10306   10307#  10309   10310#  10312
10313#  10315   10316#  10318   10319#  10321   10322#  10324   10325#  10327   10328#  10330   10331#
10333   10334#  10336   10337#  10339   10340#  10342   10343#  10345   10346#  10348   10349#  10351
10352#  10354   10355#  10356   10358   10359#  10361   10362#  10364   10365#  10367   10368#  10370
10389   10390#  10392   10393#  10395   10396#  10398   10400#  10402   10406#  10408   10413#  10415
10419#  10421   10425#  10427   10428#  10430   10431#  10433   10440#  10442   10452#  10454   10455#
10457   10459#  10460   10461#  10462   10464#  10466   10467#  10469   10471#  10473   10477#  10479
10484#  10486   10490#  10492   10494#  10498   10511#  10513   10546#  10548   10549#  10551   10552#
10554   10555#  10557   10558#  10560   10561#  10563   10564#  10566   10570#  10572   10573#  10575
10576#  10578   10579#  10581   10582#  10584   10585#  10587   10588#  10590   10591#  10593   10594#
10596   10597#  10599   10600#  10602   10603#  10605   10606#  10608   10609#  10611   10612#  10614
10615#  10617   10618#  10620   10621#  10623   10624#  10626   10627#  10629   10630#  10632   10633#
10635   10636#  10638   10639#  10641   10642#  10644   10645#  10647   10648#  10650   10651#  10653
10654#  10656   10657#  10659   10660#  10662   10663#  10665   10666#  10668   10669#  10671   10673#
10675   10677#  10679   10683#  10685   10698#  10700   10701#  10703   10704#  10706   10707#  10709
10710#  10712   10713#  10715   10717#  10719   10723#  10725   10730#  10732   10736#  10738   10742#
10744   10745#  10747   10748#  10750   10757#  10759   10769#  10771   10772#  10774   10775#  10777
10778#  10780   10781#  10783   10784#  10786   10788#  10790   10794#  10796   10801#  10803   10807#
10809   10813#  10815   10828#  10830   10863#  10865   10866#  10868   10869#  10871   10872#  10874
10875#  10877   10878#  10880   10881#  10883   10887#  10890#  10892   10893#  10895   10896#
10898   10899#  10901   10902#  10904   10905#  10907   10908#  10910   10911#  10913   10914#  10916
10917#  10919   10920#  10922   10923#  10925   10926#  10928   10929#  10931   10932#  10934   10935#
10937   10938#  10940   10941#  10943   10944#  10946   10947#  10949   10950#  10952   10953#  10955
10956#  10958   10959#  10961   10962#  10964   10965#  10967   10968#  10970   10971#  10973   10974#
10976   10977#  10979   10980#  10982   10983#  10985   10986#  10988   10990#  10992   10994#  10996
11000#  11002   11015#  11017   11018#  11020   11021#  11023   11024#  11026   11027#  11029   11030#
11032   11034#  11036   11040#  11042   11047#  11049   11053#  11055   11059#  11061   11062#  11064
11065#  11067   11074#  11076   11086#  11088   11089#  11091   11092#  11094   11095#  11097   11098#
11100   11101#  11103   11105#  11107   11111#  11113   11118#  11120   11124#  11126   11130#  11132
11145#  11147   11180#  11182   11183#  11185   11186#  11188   11189#  11191   11192#  11194   11195#
11197   11198#  11200   11204#  11206   11207#  11209   11210#  11212   11213#  11215   11216#  11218
11219#  11221   11222#  11224   11225#  11227   11228#  11230   11231#  11233   11234#  11236   11237#
```

```
11239  11240#  11242  11243#  11245  11246#  11248  11249#  11251  11252#  11254  11256#  11257
11258#  11260  11261#  11263  11264#  11266  11267#  11269  11270#  11272  11273#  11275  11276#
11278  11279#  11281  11282#  11284  11285#  11287  11288#  11290  11291#  11293  11294#  11296
11297#  11299  11300#  11302  11303#  11305  11307#  11309  11311#  11313  11317#  11319  11322#
11334  11336#  11337  11338#  11340  11341#  11343  11344#  11346  11347#  11349  11351#  11353
11357#  11359  11364#  11366  11370#  11372  11376#  11378  11379#  11381  11382#  11384  11391#
11399  11403#  11405  11406#  11408  11409#  11411  11412#  11414  11415#  11417  11418#  11420
11422#  11424  11428#  11430  11435#  11437  11441#  11443  11447#  11449  11462#  11464  11497#
11499  11500#  11502  11503#  11505  11506#  11508  11509#  11511  11512#  11514  11515#  11517
11521#  11523  11524#  11526  11527#  11529  11530#  11532  11533#  11536  11536#  11538  11539#
11541  11542#  11544  11545#  11547  11548#  11550  11551#  11553  11554#  11556  11557#  11559
11560#  11562  11563  11564#  11566  11567#  11569  11571  11572#  11574  11575#  11577  11578#
11580  11581#  11583  11584#  11586  11587#  11589  11590#  11592  11593#  11595  11596#  11598
11599#  11601  11602#  11604  11605#  11607  11608#  11610  11611#  11613  11614#  11616  11617#
11619  11620#  11622  11623#  11625  11626#  11628  11630  11634#  11636  11649#  11651  11652#  11654
11655#  11657  11658#  11660  11661#  11663  11664#  11666  11668#  11670  11674#  11676  11681#
11683  11687#  11689  11690#  11693  11696#  11698  11699#  11701  11708#  11710  11720#  11722
11723#  11726  11728#  11729  11730  11731  11732#  11734  11735#  11737  11739#  11741  11745#
11747  11750#  11751  11753#  11760  11764#  11766  11779#  11781  11814#  11816  11817#  11819
11820#  11822  11823#  11825  11826  11828  11829  11831  11832#  11834  11838#  11840  11841#
11843  11844#  11846  11847#  11849  11850#  11852  11853#  11855  11856#  11858  11859#  11861
11862#  11864  11865#  11867  11868  11870  11871#  11873  11874#  11876  11877#  11879  11880#
11882  11883#  11885  11886#  11888  11890#  11891  11892#  11894  11895#  11897  11898#  11900
11901#  11903  11904#  11906  11907#  11909  11910#  11912  11913#  11915  11916#  11918  11919#
11921  11922#  11924  11925#  11927  11928#  11930  11931#  11933  11934#  11936  11937#  11939
11941#  11943  11945#  11947  11951#  11953  11956#  11960  11969#  11971  11972#  11974  11975#
11977  11978#  11980  11981#  11983  11985#  11987  11991#  11993  11998#  12000  12004#  12006
12010#  12012  12013#  12015  12016#  12018  12025#  12027  12037#  12039  12040#  12042  12043#
12045  12046#  12048  12049#  12051  12054  12056#  12058  12062#  12064  12069#  12071
12075#  12077  12081#  12083  12085  12131#  12133  12134#  12136  12137#  12139  12140#
12142  12143#  12145  12146#  12148  12149#  12151  12155#  12157  12158#  12160  12161#  12163
12164#  12166  12167#  12168  12170#  12172  12173#  12175  12176#  12178  12179#  12181  12182#
12184  12185#  12187  12188  12190  12191#  12193  12194#  12196  12197#  12199  12200#  12202
12203#  12205  12206#  12208  12209#  12211  12212#  12214  12215#  12217  12218#  12220  12221#
12223  12224#  12226  12227#  12229  12230#  12232  12233#  12235  12236#  12238  12239#  12241
12242#  12244  12245#  12247  12248#  12250  12251#  12253  12254#  12256  12258#  12260  12262#
12264  12268#  12270  12272#  12274  12275#  12277  12279#  12281  12282#  12284  12285#  12287
12288#  12290  12291#  12293  12294  12296#  12310  12311#  12313  12317  12318#  12320  12321#
12323  12324#  12326  12327#  12329  12330#  12332  12333#  12335  12336#  12338  12339#  12365
12366#  12368  12369#  12371  12372#  12374  12375  12376#  12378  12379#  12381  12382#  12384  12398#
12400  12413#  12415  12416#  12418  12419#  12421  12422#  12424  12425#  12427  12428#  12430  12462
12462#  12465  12466#  12468  12469#  12471  12472#  12474  12475#  12477  12478#  12480  12481#  12483  12494#
12496  12497#  12499  12500#  12502  12503  12504  12505#  12507  12508#  12510  12511#  12513  12514#  12516  12517#  12519  12520#  12522  12523#
12525  12526#  12528  12529#  12531  12532#  12534  12535#  12537  12538#  12540  12541#  12543  12543#
12544#  12546  12547#  12549  12550#  12552  12553#  12555  12556#  12558  12559#  12561  12562#
12564  12565#  12567  12568#  12570  12571#  12573  12575#  12577  12579#  12581  12585#  12587
12600#  12602  12603#  12605  12606#  12608  12609#  12611  12612#  12614  12615#  12617  12619#
12621  12625#  12627  12632#  12634  12638#  12640  12644#  12646  12647#  12649  12650#  12652
12659#  12661  12671#  12673  12674#  12676  12677#  12679  12680#  12682  12683#  12685  12686#
12688  12690#  12692  12696#  12698  12703#  12705  12709#  12711  12715#  12717  12730#  12732
12765#  12767  12768#  12770  12771#  12773  12774#  12776  12777#  12779  12780#  12782  12783#
12785  12789#  12791  12792#  12794  12795#  12797  12798#  12800  12801#  12803  12804#  12806
12807#  12809  12810#  12812  12813#  12815  12816#  12818  12819#  12821  12822#  12824  12825#
12827  12828#  12830  12831#  12833  12834#  12836  12837#  12839  12840#  12842  12843#  12845
12846#  12848  12849#  12851  12852#  12854  12855#  12857  12858#  12860  12861#  12863  12864#
```



```
                    14656  14664#  14666  14678#  14701  14702#  14705#  14707  14708#  14710  14711#  14713
                    14714#  14716  14717#  14723  14724#  14725  14726#  14728  14732#  14734  14738#  14740  14747#
                    14749  14750#  14752  14758#  14760  14764#  14786  14815#  14817  14818#  14820  14821#  14823
                    14824#  14826  14830#  14832  14833#  14836  14839#  14841  14846#  14847  14851#  14853  14860#
                    14862  14866#  14868  14869  14871  14872#  14874  14891#  14893  14894#  14896  14897#  14899
                    14900#  14902  14903#  14905  14909#  14911  14915#  14917  14921#  14923  14930#  14932  14936#
                    14938  14939#  14941  14942#  14944  14945#  14957  14958#  14960  14961#  14963  14964#  14982
                    14987#  14989  14993#  14995  14979#  14981  14985#  14987  14988#  14990  14994#  14996  15010#
                    15012  15013#  15015  15016#  15018  15019#  15021  15022#  15024  15025#  15027  15028#  15030
                    15034#  15036  15041#  15043  15047#  15049
$40CAT= ###### U     1046
.       = 037470     203#  205  208#  221  222#  224#  226#  228#  232#  236#  246#  294  380#
                     381#  382#  383#  392#  480#  486  487#  489#  491#  534#  536#  537#  538#
                     539#  541#  542#  543#  544#  546#  547#  548#  549#  551#  552#  553#  554#
                     556#  557#  558#  559#  561#  562#  563#  564#  566#  567#  568#  569#  571#
                     572#  573#  574#  576#  577#  578#  579#  581#  582#  583#  584#  586#  587#
                     588#  589#  591#  592#  593#  594#  596#  597#  598#  599#  601#  602#  603#
                     604#  606#  607#  608#  609#  611#  612#  613#  614#  775  785  892#  905
                    1010  1081  1082  1174  1226#  1236  1328#  1329  1335#  1389  1678  1707  1750#
                    1769#  1785#  1787#  1841  1844  1865  1964  2005  2088  2092  2138  2169  2202
                    2205  2281#  2478#  2675#  2872#  3070#  3267#  3520#  3717#  3914#  4111#  4308#  4505#
                    4702#  4899#  5097#  5274#  5481#  5680#  5856#  6102#  6309#  6516#  6723#  6930#  7137#
                    7343#  7517#  7760#  7957#  8137#  8294#  8455#  8660#  8834#  8960#  9277#  9594#  9911#
                    10228#  10545#  10862#  11179#  11496#  11813#  12130#  12447#  12764#  13081#  13398#  13713#  13856#
                    14480#  14698#  14808#  15310#  15312#
.ADC    = 000100     1#  5162  5232  8009  8097  8238  8398  8533  8617  8696  8783  8865  8916
                    12905  12973  13792  13968  14770  14773
.ADD    = 000000     1#  9095  9166  9231  9412  9483  9548  9729  9800  9865  10046  10117  10182
                    10363  10434  10499  10680  10686  10751  10754  10816  10997  11068  11133  11314  11385
                    11450  11631  11702  11767  11948  12019  12084  12265  12336  12401  12582  12653  12718
                    12899  12970  13035  13216  13287  13352  13533  13604  13669  14863  14933
.ADDWC= 000020       1#  11637  11705
.ADVAN  006050      1429#  1569
.AND    = 000260     1#  3092  3181  3309  3407  7861  9735  9803  14535  14836  14906  14970  14991
.B80    = 002000     1#  7410  7617  7644  7827  7988  8871  8922  14571  14854  15037
.B81    = 002400     1#  14577  14592  14924
.B84    = 003000     1#  13732  13760  13786  14559  14741
.B87    = 003400     1#  2362  2442  2659  2639  2756  2836  2953  3033  3160  3249  3386  3484
                    3601  3681  3798  3878  3995  4075  4192  4272  4389  4469  4586  4666  4783
                    4863  4980  5060  5168  5238  5360  5445  5567  5652  5774  5859  5981  6066
                    6188  6273  6395  6480  6602  6687  6809  6894  7016  7101  7223  7308  8232
                    8389  13798  14547  14735  14761  14848  14857  14918  14927  14982  14997  15031
.BC     = 001000     1#  7833  8250  8410  8545  8629  8705  8795  13983  14015  14342  14776
.BEGIN  003464      920#
.BR     = 000400     1#  2324  2340  2347  2353  2365  2414  2420  2427  2433  2450  2453  2531
                    2537  2544  2550  2562  2611  2617  2624  2630  2647  2650  2728  2734  2741
                    2747  2759  2808  2814  2821  2827  2844  2847  2905  2931  2938  2944  2956
                    3005  3011  3018  3024  3041  3044  3132  3138  3145  3151  3163  3221  3227
                    3234  3240  3257  3260  3358  3364  3371  3377  3389  3456  3462  3469  3475
                    3492  3498  3573  3579  3586  3592  3604  3653  3659  3666  3672  3689  3692
                    3770  3776  3783  3789  3801  3850  3856  3863  3869  3886  3889  3967  3973
                    3980  3986  3998  4047  4053  4060  4066  4083  4086  4164  4170  4177  4183
                    4195  4244  4250  4257  4263  4280  4283  4361  4367  4374  4380  4392  4441
                    4447  4454  4460  4477  4480  4558  4564  4571  4577  4589  4638  4644  4651
                    4657  4674  4677  4755  4761  4768  4774  4786  4835  4841  4848  4854  4871
                    4874  4952  4958  4965  4971  4983  5032  5038  5045  5051  5068  5071  5140
```

```
5143  5150  5153  5171  5210  5213  5220  5223  5246  5249  5332  5338  5345
5351  5363  5417  5423  5430  5436  5453  5456  5539  5545  5552  5558  5570
5624  5630  5637  5643  5660  5663  5746  5752  5759  5765  5777  5831  5837
5944  5950  5957  5970  5953  5959  5966  5972  5984  6038  6044  6051  6057
6074  6077  6160  6166  6173  6179  6191  6245  6251  6258  6264  6281  6284
6367  6373  6380  6386  6398  6452  6458  6465  6471  6488  6491  6574  6580
6587  6593  6605  6659  6665  6672  6678  6695  6698  6781  6787  6794  6800
6812  6866  6872  6879  6885  6902  6905  6988  6994  7001  7007  7019  7073
7079  7086  7092  7109  7112  7195  7201  7208  7214  7226  7280  7286  7293
7299  7316  7319  7444  7447  7478  7481  7488  7491  7581  7604  7721  7724
7731  7734  7836  7849  7852  7868  7871  7958  8012  8079  8082  8100  8107
8110  8204  8210  8217  8223  8260  8267  8270  8361  8367  8374  8380  8392
8420  8427  8430  8514  8517  8524  8527  8548  8598  8601  8608  8611  8632
8640  8643  8708  8758  8764  8771  8777  8798  8806  8809  8885  8888  8895
8898  8929  8932  9135  9141  9148  9154  9206  9212  9219  9223  9263  9346
9349  9452  9458  9465  9471  9523  9529  9536  9542  9556  9563  9566  9769
9775  9782  9788  9840  9846  9853  9859  9873  9880  9883  10086 10092 10099
10105 10157 10163 10170 10176 10190 10197 10200 10403 10409 10416 10422 10474
10480 10487 10493 10507 10514 10517 10720 10726 10733 10739 10791 10797 10804
10810 10824 10831 10834 11037 11043 11050 11056 11108 11114 11121 11127 11141
11148 11151 11354 11360 11367 11373 11425 11431 11438 11444 11458 11465 11468
11671 11677 11684 11690 11742 11748 11755 11761 11775 11782 11735 11988 11994
12001 12007 12059 12065 12072 12078 12092 12099 12102 12305 12311 12318 12324
12376 12382 12389 12395 12409 12416 12419 12422 12428 12435 12441 12693 12699
12706 12712 12726 12733 12736 12939 12945 12955 12958 13010 13016 13023 13029
13043 13050 13053 13056 13062 13069 13275 13327 13333 13340 13346 13360 13367
13370 13373 13379 13386 13392 13644 13650 13657 13663 13677 13684 13687 13745
13748 13763 13777 13780 13801 13815 13818 13825 13828 13986 13993 13999 14018
14024 14040 14056 14066 14076 14086 14096 14106 14116 14128 14138 14148 14158
14168 14178 14188 14198 14214 14233 14246 14259 14272 14285 14298 14311 14324
14337 14383 14386 14395 14404 14411 14419 14451 14455 14586 14601 14619 14657
14660 14667 14670 14764 14779 14787 14875 14945 15000 15044 15050
:BSBRG= 160000  1# 14956 14976 14986 15001
:BSIMM= 100000  1# 2310 2335 2341 2348 2354 2363 2366 2390 2415 2421 2428 2434
2443 2451 2454 2507 2532 2538 2545 2551 2560 2563 2587 2612 2618
2625 2631 2640 2648 2651 2704 2729 2735 2742 2748 2757 2760 2784
2809 2815 2822 2828 2837 2845 2848 2901 2926 2932 2939 2945 2954
2957 2981 3006 3012 3019 3025 3034 3042 3045 3108 3133 3139 3146
3152 3161 3164 3197 3222 3228 3235 3241 3250 3258 3261 3334 3359
3365 3372 3378 3387 3390 3432 3435 3463 3470 3476 3485 3493 3498
3549 3574 3580 3587 3593 3612 3615 3622 3654 3660 3667 3673 3682
3690 3693 3746 3771 3777 3784 3790 3799 3802 3826 3851 3857 3864
3870 3879 3887 3890 3943 3949 3974 3981 3987 3996 3999 4023 4048
4054 4061 4067 4076 4084 4087 4140 4165 4171 4178 4184 4193 4196
4220 4245 4251 4258 4264 4273 4281 4284 4337 4362 4368 4375 4381
4390 4393 4417 4442 4448 4455 4461 4470 4479 4481 4534 4559 4565
4572 4578 4587 4590 4614 4636 4645 4652 4658 4667 4675 4678 4731
4756 4762 4769 4775 4784 4787 4811 4814 4842 4849 4855 4864 4872
4875 4928 4953 4959 4966 4972 4981 4984 5008 5033 5039 5046 5052
5061 5069 5072 5125 5141 5144 5151 5154 5169 5172 5195 5211 5214
5221 5224 5239 5247 5250 5305 5333 5339 5346 5352 5361 5364 5390
5418 5424 5431 5437 5446 5454 5457 5512 5540 5546 5553 5559 5568
5571 5597 5625 5631 5638 5644 5653 5661 5664 5719 5747 5753 5760
5766 5775 5778 5804 5832 5838 5845 5851 5860 5868 5871 5967 5954
5960 5967 5973 5982 5985 6011 6039 6045 6052 6058 6067 6075 6078
6133 6161 6167 6174 6180 6189 6192 6218 6246 6252 6259 6265 6274
```

```
                6282  6285  6340  6358  6374  6381  6387  6396  6399  6425  6453  6459  6466
                6472  6481  6489  6492  6547  6575  6581  6588  6594  6603  6606  6632  6660
                6666  6673  6679  6688  6696  6699  6754  6782  6788  6795  6801  6810  6813
                6839  6867  6873  6880  6886  6895  6903  6905  6961  6969  6975  7002  7008
                7017  7020  7046  7074  7080  7087  7093  7102  7110  7113  7168  7196  7202
                7209  7215  7224  7227  7263  7281  7287  7294  7300  7309  7317  7320  7411
                7423  7445  7448  7460  7479  7482  7489  7442  7618  7645  7660  7682  7685
                7700  7722  7725  7736  7756  7828  7834  7837  7850  7863  7869  7872  7956
                7959  7985  8001  8013  8058  8080  8083  8089  8101  8108  8111  8180  8205
                8211  8218  8224  8233  8251  8261  8268  8271  8337  8362  8368  8375  8381
                8340  8343  8411  8421  8428  8431  8487  8515  8518  8525  8528  8546  8549
                8577  8593  8602  8609  8612  8630  8633  8641  8644  8706  8709  8737  8759
                8765  8772  8778  8796  8799  8807  8810  8872  8886  8889  8896  8899  8923
                8930  8933  9111  9136  9142  9149  9155  9182  9207  9213  9220  9226  9237
                9240  9247  9250  9256  9262  9268  9272  9276  9279  9282  9286  9290  9293
                9554  9557  9564  9567  9745  9770  9776  9783  9789  9816  9841  9847  9854
                9860  9871  9874  9881  9904  10062  10087  10093  10100  10106  10133  10158  10164
                10171  10177  10180  10191  10198  10201  10379  10404  10410  10417  10423  10450  10475
                10484  10488  10494  10505  10508  10515  10518  10526  10721  10727  10734  10740  10767
                10792  10798  10805  10811  10822  10825  10832  10835  11013  11038  11044  11051  11057
                11084  11109  11115  11132  11138  11139  11145  11149  11152  11330  11355  11361  11368
                11374  11401  11406  11432  11438  11445  11456  11459  11466  11469  11647  11672  11678
                11685  11691  11718  11743  11749  11756  11762  11773  11776  11783  11786  11964  11989
                11995  12002  12008  12036  12039  12046  12073  12079  12090  12093  12100  12103  12281
                12306  12312  12319  12346  12349  12377  12383  12390  12396  12407  12410  12417  12420
                12598  12623  12629  12636  12642  12669  12694  12700  12707  12713  12724  12727  12734
                12737  12915  12940  12956  12963  12969  12996  13011  13017  13024  13030  13041  13044
                13051  13054  13222  13257  13263  13270  13276  13303  13329  13334  13341  13347  13358
                13361  13368  13374  13399  13424  13480  13487  13493  13520  13545  13651  13658  13664
                13675  13678  13686  13708  13723  13746  13749  13761  13754  13778  13781  13787  13799
                13802  13816  13819  13825  13832  13884  13887  13894  14000  14007  14016  14019  14025
                14041  14057  14067  14077  14087  14097  14107  14117  14129  14139  14149  14159  14169
                14179  14189  14199  14215  14234  14247  14260  14273  14286  14299  14312  14325  14338
                14365  14384  14387  14393  14396  14402  14405  14412  14415  14430  14452  14548  14560
                14572  14578  14587  14593  14602  14617  14620  14636  14658  14661  14668  14671  14736
                14742  14762  14765  14777  14780  14788  14849  14855  14858  14919  14925  14928  14983
.BSMEM= 140000    1#
.BZ   = 001400    1#  2309  2389  2506  2586  2703  2783  2900  2980  3107  3196  3333  3431
                3548  3628  3745  3825  3942  4022  4139  4219  4336  4416  4533  4613  4730
                4810  4927  5007  5124  5194  5304  5389  5511  5536  5718  5803  5925  6010
                6132  6217  6339  6424  6546  6631  6753  6838  6960  7045  7167  7252  7422
                7469  7569  7639  7655  8000  8067  8080  8129  8336  8406  8576  8736  9110
                9181  9236  9427  9498  9553  9744  9815  9870  10061  10132  10187  10378  10449
                10504  10686  10766  10821  11012  11083  11138  11329  11400  11455  11646  11717  11772
                11963  12034  12089  12280  12351  12406  12597  12668  12723  12914  12985  13040  13231
                13302  13357  13548  13619  13674  14006  14364  14401  14429  14616  14635
.CMVRT  006160  1462#  1568
.CONVR  006154  1461#  1567
.CO   = 000400    1#
.DATAC  007420  1564  1741#
.DBR  = 000400    1#  2283  2290  2306  2331  2337  2344  2350  2356  2370  2386  2411  2417
                2424  2430  2436  2447  2480  2487  2503  2528  2534  2541  2547  2553  2567
                2583  2608  2614  2621  2627  2633  2644  2677  2684  2700  2725  2731  2738
                2744  2750  2764  2780  2805  2811  2818  2824  2830  2841  2874  2881  2897
                2922  2928  2935  2941  2947  2961  2977  3002  3008  3015  3021  3027  3038
```

3072	3079	3092	3104	3129	3135	3142	3148	3154	3168	3181	3193	3218
3224	3231	3237	3243	3254	3260	3266	3309	3324	3330	3355	3361	3368
3374	3380	3394	3407	3422	3428	3453	3459	3466	3472	3478	3489	3522
3529	3545	3570	3576	3583	3589	3595	3609	3625	3650	3656	3663	3669
3675	3686	3719	3726	3745	3767	3773	3780	3786	3792	3806	3822	3847
3853	3860	3866	3872	3883	3916	3923	3939	3964	3970	3977	3983	3989
4003	4019	4044	4050	4057	4063	4069	4080	4113	4120	4136	4161	4167
4174	4180	4186	4200	4216	4241	4247	4254	4260	4266	4277	4310	4317
4333	4358	4364	4371	4377	4383	4397	4413	4438	4444	4451	4457	4463
4474	4507	4514	4530	4555	4561	4568	4574	4580	4594	4610	4635	4641
4648	4654	4660	4671	4704	4711	4727	4752	4758	4765	4771	4777	4791
4807	4832	4838	4845	4851	4857	4868	4901	4908	4924	4949	4955	4962
4968	4974	4980	5004	5029	5035	5042	5048	5054	5065	5099	5112	5137
5147	5156	5162	5207	5217	5226	5243	5276	5283	5301	5310	5316	5322
5329	5336	5343	5348	5354	5368	5388	5395	5401	5407	5414	5420	5427
5433	5439	5450	5483	5440	5508	5517	5523	5529	5536	5542	5549	5555
5561	5575	5583	5602	5608	5614	5621	5627	5634	5640	5646	5657	5690
5697	5715	5724	5730	5736	5743	5749	5756	5762	5768	5782	5800	5809
5815	5821	5828	5834	5841	5847	5853	5864	5897	5904	5922	5931	5937
5943	5950	5956	5963	5969	5975	5989	6007	6016	6022	6028	6035	6041
6048	6054	6060	6071	6104	6111	6129	6138	6144	6150	6157	6163	6170
6176	6182	6196	6214	6223	6229	6235	6242	6248	6255	6261	6267	6278
6311	6318	6336	6345	6351	6357	6364	6370	6377	6383	6389	6403	6421
6430	6436	6442	6449	6455	6462	6468	6474	6485	6518	6525	6543	6552
6558	6564	6571	6577	6584	6590	6596	6610	6628	6637	6643	6649	6656
6662	6669	6675	6681	6692	6725	6732	6750	6759	6765	6771	6778	6784
6791	6797	6803	6817	6835	6844	6850	6856	6863	6869	6876	6882	6888
6899	6932	6939	6957	6966	6972	6978	6985	6991	6998	7004	7010	7024
7042	7051	7057	7063	7070	7076	7083	7089	7095	7106	7139	7146	7164
7173	7179	7185	7192	7198	7205	7211	7217	7231	7249	7258	7264	7270
7277	7283	7290	7296	7302	7313	7344	7365	7377	7383	7389	7395	7401
7407	7413	7419	7428	7434	7441	7450	7456	7475	7485	7518	7554	7557
7569	7575	7581	7587	7593	7599	7605	7614	7623	7629	7635	7641	7647
7656	7665	7678	7687	7696	7705	7718	7728	7761	7797	7803	7818	7824
7846	7855	7861	7865	7904	7925	7931	7937	7967	7973	7979	7985	8015
8063	8076	8104	8139	8145	8176	8188	8194	8201	8207	8214	8220	8226
8264	8296	8302	8333	8345	8351	8358	8364	8371	8377	8383	8424	8456
8477	8483	8492	8498	8504	8511	8521	8536	8542	8552	8567	8573	8582
8588	8595	8605	8620	8636	8637	8669	8687	8702	8712	8727	8733	8742
8748	8755	8768	8786	8792	8803	8835	8868	8882	8892	8919	8926	8967
9092	9095	9119	9125	9132	9145	9163	9166	9190	9196	9203	9216	9228
9243	9284	9409	9412	9436	9442	9449	9462	9480	9483	9507	9513	9520
9533	9545	9560	9601	9726	9729	9753	9759	9766	9779	9797	9800	9824
9830	9837	9850	9862	9877	9918	10043	10046	10070	10076	10083	10096	10114
10117	10141	10147	10154	10167	10179	10194	10235	10360	10363	10387	10393	10400
10413	10431	10434	10458	10464	10471	10484	10496	10511	10552	10677	10680	10704
10710	10717	10730	10748	10751	10775	10781	10788	10801	10813	10828	10869	10994
10997	11021	11027	11034	11047	11065	11068	11092	11098	11105	11118	11130	11145
11186	11311	11314	11338	11344	11351	11364	11382	11385	11409	11415	11422	11435
11447	11462	11503	11628	11631	11655	11661	11668	11681	11699	11702	11726	11732
11739	11752	11764	11779	11820	11945	11948	11972	11978	11985	11998	12016	12019
12043	12049	12056	12069	12081	12096	12137	12262	12265	12289	12295	12302	12315
12333	12336	12360	12366	12373	12386	12398	12413	12454	12579	12582	12606	12612
12619	12632	12650	12653	12677	12683	12690	12703	12715	12730	12771	12896	12899
12923	12929	12936	12949	12967	12970	12994	13000	13007	13020	13032	13047	13088
13213	13216	13240	13246	13253	13266	13284	13287	13311	13317	13324	13337	13349

```
                    13364  13405  13530  13533  13557  13563  13570  13583  13601  13604  13628  13634  13641
                    13654  13666  13681  13714  13729  13742  13751  13757  13774  13783  13795  13812  13822
                    13857  13974  13980  13980  13988  14021  14027  14063  14063  14073  14083  14093  14103
                    14113  14125  14136  14145  14158  14166  14175  14186  14186  14204  14211  14217  14223
                    14230  14236  14243  14248  14258  14262  14267  14275  14286  14288  14295  14301  14308
                    14314  14321  14327  14334  14346  14351  14373  14380  14408  14420  14426  14435  14441
                    14448  14481  14511  14517  14528  14538  14544  14550  14556  14562  14568  14574  14589
                    14607  14613  14626  14632  14641  14647  14654  14664  14698  14714  14723  14732  14738
                    14747  14758  14784  14821  14833  14839  14845  14851  14860  14866  14894  14903  14909
                    14915  14921  14930  14936  14956  14967  14973  14979  14985  14994  15010  15016  15022
                    15028  15034  15041  15047
.DBRSH= 001400        1#  2360  2440  2557  2637  2754  2834  2951  3031  3158  3247  3384  3482
                    3599  3679  3796  3876  3993  4073  4190  4270  4387  4467  4584  4664  4781
                    4861  4978  5058  5175  5255  5372  5452  5569  5650  5772  5857  5979  6064  6186  6271
                    6393  6478  6600  6685  6807  6892  7014  7099  7221  7306  7372  7564  7950
                    8230  8387
.DBRSP= 003400        1#  9101  9169  9418  9486  9735  9803  10052  10120  10369  10437  10686  10754
                    11003  11071  11320  11388  11637  11705  11954  12022  12271  12339  12588  12656  12905
                    12973  13222  13290  13558  13607
.DD   = 003000        2306#  2386#  2503#  2583#  2700#  2780#  2897#  2977#  3094#  3104#  3181#  3193#  3309#
                    3324#  3330#  3407#  3422#  3439#  3456#  3473#  3522#  3539#  3556#  4019#  4136#  4216#
                    4333#  4413#  4530#  4610#  4727#  4807#  4924#  5004#  5121#  5191#  5301#  5386#  5508#
                    5593#  5715#  5800#  5922#  6007#  6129#  6214#  6336#  6421#  6543#  6628#  6750#  6835#
                    6957#  7042#  7164#  7249#  7419#  7456#  7541#  7611#  7658#  7696#  7815#  7861#  8054#
                    8176#  8247#  8333#  8407#  8483#  8560#  8573#  8645#  8702#  8732#  8792#  8964#  9076#  9096
                    9107#  9166#  9167  9170#  9231#  9312#  9413  9416#  9424#  9483#  9484  9486#  9495#
                    9548#  9729#  9730  9735#  9741#  9800#  9801  9803#  9812#  9865#  10046#  10047  10052#
                    10058#  10117#  10118  10120#  10124#  10183#  10184  10238#  10375#  10434#  10435  10437#
                    10446#  10499#  10680#  10681  10686#  10742#  10752  10754#  10763#  10816#  10997#  10998
                    11009#  11064#  11069  11080#  11134#  11314#  11315  11320#  11326#  11386  11388#  11397#
                    11450#  11631#  11632  11637#  11643#  11702#  11703  11705#  11714#  11767#  11948#  11949  11954#
                    11960#  12019#  12020  12022#  12031#  12090#  12091  12096  12277#  12336#  12337  12348#  12401#
                    12582#  12643  12644#  12653#  12654  12713#  12714  12716#  12900  12911#  12970#  12971  12982#
                    13035#  13216#  13217  13222#  13280#  13287#  13288  13340  13469#  13528#  13533#  13534  13545#
                    13604#  13605  13616#  13659#  13660#  14461#  14462  14536#  14541#  14613#  14632#  14720#  14729#
                    14753#  14827#  14836#  14842#  14853#  14906#  14912#  14933#  14970#  14976#  14991#
.DEC  = 000160        1#  7952  7997  8085  8901  8913  8982  9233  9299  9550  9616  9867  9933
                    10184  10250  10501  10567  10818  10884  11135  11201  11452  11518  11763  11835  12086
                    12152  12403  12469  12720  12786  13037  13103  13354  13420  13539  13607  13671  14003
                    14037  14388
.DELAY  007304      1562  1714#
.DMEM = 002400        1#  2318  2324  2356  2404  2515  2521  2595  2601  2712  2718  2792  2798
                    2909  2915  2989  2995  3086  3116  3122  3175  3205  3211  3303  3318  3342
                    3348  3401  3416  3440  3446  3557  3563  3637  3643  3754  3760  3834  3840
                    3951  3957  4031  4037  4148  4154  4228  4234  4345  4351  4425  4431  4542
                    4548  4622  4628  4739  4745  4819  4825  4936  4942  5016  5022  5106  5121
                    5165  5176  5191  5236  5289  5356  5374  5380  5496  5502  5581  5587  5703
                    5709  5788  5794  5910  5916  5985  6001  6117  6123  6202  6208  6324  6330
                    6409  6415  6531  6537  6616  6622  6738  6744  6823  6829  6945  6951  7030
                    7036  7152  7158  7237  7243  7468  7671  7711  7809  7839  7946  7931  7994
                    8054  8069  8154  8164  8241  8247  8311  8321  8401  8407  8474  8490  8856
                    8875  8904  8988  8991  8994  8997  9000  9003  9006  9009  9012  9015  9018
                    9021  9024  9027  9030  9033  9036  9039  9042  9045  9048  9051  9054  9057
                    9060  9063  9066  9069  9072  9075  9078  9081  9107  9178  9305  9308  9311
                    9314  9317  9320  9323  9326  9329  9332  9335  9338  9341  9344  9347  9350
                    9353  9356  9359  9362  9365  9368  9371  9374  9377  9380  9383  9386  9389
```

```
                9382  9385  9388  9424  9485  9622  9625  9628  9631  9634  9637  9640  9643
                9646  9649  9652  9655  9658  9661  9664  9667  9670  9673  9676  9679  9682
                9685  9688  9691  9694  9697  9700  9703  9706  9709  9712  9715  9741  9812
                9969  9972  9975  9978  9981  9984  9987  9990  9993  9996  9999 10002 10005 10008 10011 10014
               10017 10020 10023 10026 10029 10032 10035 10038 10041 10044 10047 10050 10053 10056 10059 10062
               10271 10274 10277 10280 10283 10286 10289 10292 10295 10298 10301 10304 10307 10310 10313 10316
               10319 10322 10325 10328 10331 10334 10337 10340 10343 10346 10349 10375 10378 10381 10384 10387
               10390 10393 10396 10399 10402 10405 10408 10411 10414 10417 10420 10423 10426 10429 10432 10435
               10438 10441 10444 10447 10450 10453 10456 10459 10462 10465 10468 10471 10474 10477 10480 10483
               10486 10489 10492 10495 10498 10501 10504 10507 10510 10513 10516 10519 10522 10525 10528 10531
               10534 10537 10540 10543 10546 10549 10552 10555 10558 10561 10564 10567 10570 10573 10576 10579
               10582 10585 10588 10591 10594 10597 10600 10603 10606 10609 10612 10615 10618 10621 10624 10627
               10630 10633 10636 10639 10642 10645 10648 10651 10654 10657 10660 10663 10666 10669 10672 10675
               10678 10681 10684 10687 10690 10693 10696 10699 10702 10705 10708 10711 10714 10717 10720 10723
               10726 10729 10732 10735 10738 10741 10744 10747 10750 10753 10756 10759 10762 10765 10768 10771
               10774 10777 10780 10783 10787 10791 10795 10799 10803 10807 10811 10815 10819 10823 10827 10831
               10835 10839 10843 10847 10851 10855 10859 10863 10867 10871 10875 10879 10883 10887 10891 10895
               10899 10903 10907 10911 10915 10919 10923 10927 10931 10935 10939 10943 10947 10951 10955 10959
               10963 10967 10971 10974 10977 10980 10983 10987 10991 10995 10999 11003 11007 11011 11015 11019
               11023 11027 11031 11035 11039 11043 11047 11051 11055 11059 11063 11067 11071 11075 11079 11083
               11087 11091 11095 11099 11103 11107 11111 11115 11119 11123 11127 11131 11135 11139
```

```
                    4231  4237  4339  4342  4348  4354  4419  4422  4428  4434  4536  4539  4545
                    4551  4616  4619  4625  4631  4733  4736  4742  4748  4813  4816  4822  4828
                    4930  4933  4939  4945  5010  5013  5019  5025  5127  5130  5133  5197  5200
                    5203  5307  5313  5319  5325  5382  5388  5404  5410  5514  5520  5526  5532
                    5599  5605  5611  5617  5721  5727  5733  5739  5806  5812  5818  5824  5828
                    5934  5940  5946  6013  6019  6025  6031  6135  6141  6147  6153  6220  6226
                    6232  6238  6343  6348  6354  6360  6427  6433  6439  6445  6549  6555  6561
                    6567  6634  6640  6646  6653  6756  6762  6768  6774  6841  6847  6853  6859
                    6963  6969  6975  6981  7048  7054  7060  7066  7170  7176  7182  7188  7255
                    7261  7267  7273  7363  7404  7425  7431  7437  7462  7465  7471  7548  7554
                    7611  7638  7662  7668  7674  7702  7708  7714  7794  7815  7821  7832  7861
                    7922  7982  8021  8060  8066  8072  8182  8185  8191  8197  8339  8342  8348
                    8354  8489  8495  8501  8507  8575  8585  8591  8739  8745  8751  8878  9113
                    9116  9122  9128  9184  9187  9193  9199  9430  9433  9439  9445  9501  9504
                    9510  9516  9747  9750  9756  9762  9818  9821  9827  9833  10064 10067 10073
                   10079 10135 10138 10144 10150 10381 10384 10390 10396 10452 10455 10461 10467
                   10698 10701 10707 10713 10769 10772 10778 10784 11015 11018 11024 11030 11086
                   11089 11095 11101 11332 11335 11341 11347 11403 11406 11412 11418 11649 11652
                   11658 11664 11720 11723 11729 11736 11966 11969 11975 11981 12037 12040 12046
                   12052 12283 12286 12292 12298 12354 12357 12363 12369 12600 12603 12609 12615
                   12671 12674 12680 12686 12817 12820 12826 12832 12888 12891 12897 13003 13324
                   13237 13243 13249 13305 13308 13314 13320 13351 13354 13360 13366 13622 13625
                   13631 13637 13738 13754 13770 13808 13876 13879 13882 13885 13888 13982 13985
                   14367 14370 14376 14432 14438 14444 14490 14493 14526 14541 14553 14565 14638
                   14644 14650 14720 14726 14753 14815 14827 14842 14872 14897 14912 14942 14961
                   14976 14991 15013 15019 15025
.DSP  = 003000        18  2286  2297  2300  2377  2380  2483  2494  2497  2574  2577  2680  2691
                    2694  2771  2774  2877  2880  2971  2974  3075  3083  3088  3172
                    3178  3187  3292  3300  3306  3315  3321  3390  3404  3413  3419  3525  3536
                    3539  3616  3619  3722  3733  3736  3813  3816  3919  3930  3933  4010  4013
                    4116  4127  4130  4207  4210  4313  4324  4327  4404  4407  4510  4521  4524
                    4601  4604  4707  4718  4721  4788  4801  4904  4915  4918  4995  4998  5102
                    5109  5115  5160  5163  5179  5185  5230  5233  5279  5286  5292  5371  5377
                    5486  5493  5499  5578  5584  5633  5700  5706  5785  5791  5900  5907  5913
                    5992  5998  6107  6114  6120  6193  6205  6314  6321  6327  6406  6412  6521
                    6528  6534  6613  6619  6728  6735  6741  6820  6826  6935  6942  6948  7027
                    7033  7142  7149  7155  7234  7240  7247  7368  7374  7380  7386  7521  7551
                    7560  7566  7572  7578  7608  7611  7650  7690  7764  7767  7812  7815  7831
                    7858  7898  7901  7907  7938  7944  7953  7998  8004  8007  8010  8018  8024
                    8027  8033  8033  8042  8066  8072  8095  8098  8142  8157  8160  8167  8170
                    8236  8239  8298  8314  8317  8324  8327  8396  8399  8459  8468  8471  8531
                    8534  8561  8564  8615  8618  8672  8681  8684  8694  8697  8721  8724  8781
                    8784  8838  8847  8850  8853  8859  8863  8868  8902  8907  8911  8914  8917
                    8970  8973  8976  8979  8983  8986  9084  9088  9160  9172  9231  9234  9287
                    9290  9293  9296  9300  9302  9401  9415  9418  9477  9486  9489  9548  9551
                    9604  9607  9610  9613  9617  9619  9718  9721  9735  9794  9803  9806  9865
                    9868  9921  9924  9927  9930  9934  9936  10035 10049 10052 10111 10120 10123
                   10182 10185 10238 10241 10244 10247 10251 10253 10362 10366 10369 10428 10437
                   10440 10499 10502 10555 10558 10561 10564 10568 10570 10669 10683 10686 10745
                   10754 10757 10816 10819 10872 10875 10878 10881 10885 10887 10986 11000 11062
                   11074 11133 11136 11189 11192 11195 11198 11202 11204 11303 11317 11320 11379
                   11388 11391 11450 11453 11506 11509 11512 11515 11519 11521 11620 11634 11637
                   11696 11705 11708 11767 11770 11823 11826 11829 11832 11836 11838 11937 11951
                   11954 12013 12022 12025 12084 12087 12140 12143 12146 12149 12153 12155 12254
                   12268 12330 12342 12401 12404 12457 12460 12463 12466 12470 12472 12571 12585
                   12647 12659 12718 12721 12774 12777 12780 12783 12787 12789 12888 12902 12964
```

```
                    12976  13035  13038  13091  13094  13097  13100  13104  13106  13205  13219  13222  13281
                    13290  13293  13352  13355  13408  13411  13414  13417  13421  13423  13522  13536  13592
                    13610  13669  13672  13717  13720  13723  13726  13790  13793  13860  13863  13866  13927
                    13930  13933  13936  13939  13942  13945  13948  13951  13954  13957  13960  13963  13971
                    14004  14009  14013  14031  14034  14038  14046  14049  14059  14069  14079  14089  14099
                    14109  14121  14131  14141  14151  14161  14171  14181  14191  14207  14220  14226  14239
                    14252  14265  14278  14291  14304  14317  14330  14349  14362  14390  14399  14417  14508
                    14532  14536  14541  14580  14583  14595  14598  14604  14623  14702  14705  14708  14711
                    14717  14720  14726  14729  14750  14753  14766  14771  14774  14810  14813  14818  14824
                    14827  14830  14836  14842  14863  14869  14886  14889  14891  14900  14906  14912  14933
                    14939  14955  14964  14970  14976  14988  14991
.DO   = 000400        1#
.FO   = 000020        1#  2297  2315  2377  2395  2494  2512  2574  2592  2691  2709  2771  2789
                    2888  2906  2968  2986  3048  3113  3187  3202  3315  3339  3413  3437  3536
                    3554  3616  3634  3733  3751  3813  3831  3930  3948  4010  4028  4127  4145
                    4207  4225  4324  4342  4404  4422  4521  4539  4601  4619  4718  4736  4798
                    4816  4915  4933  4995  5013  7407  7413  7450  7551  7608  7614  7641  7647
                    7650  7687  7690  7812  7824  7858  7985  7991  7994  13729  13757  13783  14049
                    14059  14069  14079  14089  14099  14109  14121  14131  14141  14151  14161  14171  14181
                    14191  14502  14544  14556  14568  14574  14580  14583  14589  14595  14598  14604  14623
                    14717  14726  14732  14738  14750  14758  14784  14824  14830  14845  14851  14900  14915
                    14921  14964  14979  14988  14994  15028  15034  15047
.INC  = 000060        1#  7830  8003  8006  8091  8094  8235  8395  8530  8614  8693  8780  8862
                    8910  12271  12339  13789  13965  14012  14389  14767  14809  14812  14885  14888
.LORN = 000240        1#  9418  9486
.MINUS= 000360        1#
.MSTCL  007334      1561  17258
.MO   = 004000        1#
.OR   = 000300        1#  3324  3422  7554  7611  7815  10052  10120  14541  14720  14729  14753  14827
                    14842  14912  14976
.PLUS = 000000        1#
.RESOS  006122      1450#  1560
.ROMCL  007352      1563  17308
.SAVOS  006062      1436#  1559
.SBR  = 060000        1#  2286  2294  2300  2306  2312  2337  2350  2356  2360  2374  2380  2386
                    2392  2417  2430  2436  2440  2463  2491  2497  2503  2509  2534  2547  2553
                    2557  2571  2577  2583  2589  2614  2627  2633  2637  2680  2688  2694  2700
                    2706  2731  2744  2750  2754  2768  2774  2780  2786  2811  2824  2830  2834
                    2877  2885  2891  2897  2903  2928  2941  2947  2951  2965  2971  2977  2983
                    3008  3021  3027  3031  3075  3083  3092  3095  3104  3110  3135  3148  3154
                    3158  3172  3181  3184  3193  3199  3224  3237  3243  3247  3282  3300  3309
                    3312  3324  3330  3336  3361  3374  3380  3384  3398  3407  3410  3422  3428
                    3434  3459  3472  3478  3482  3535  3553  3539  3545  3551  3576  3589  3595
                    3599  3613  3619  3625  3631  3656  3669  3675  3679  3722  3730  3736  3742
                    3748  3773  3786  3792  3796  3810  3816  3822  3828  3853  3866  3872  3876
                    3919  3927  3933  3939  3945  3970  3983  3989  3993  4007  4013  4019  4025
                    4050  4063  4069  4073  4116  4124  4130  4136  4142  4167  4180  4186  4190
                    4204  4210  4216  4222  4247  4260  4266  4270  4313  4321  4327  4333  4339
                    4364  4377  4383  4387  4401  4407  4413  4419  4444  4457  4463  4467  4510
                    4518  4524  4530  4536  4561  4574  4580  4584  4598  4604  4610  4616  4641
                    4654  4660  4664  4707  4715  4721  4727  4733  4758  4771  4777  4781  4795
                    4801  4807  4813  4838  4861  4857  4861  4904  4912  4918  4924  4930  4955
                    4968  4974  4978  4992  4998  5004  5010  5035  5040  5054  5058  5102  5115
                    5130  5159  5162  5165  5185  5200  5229  5232  5235  5279  5288  5289  5295
                    5301  5307  5310  5313  5319  5325  5358  5371  5374  5380  5386  5392  5395
                    5398  5404  5410  5443  5486  5543  5496  5502  5508  5514  5517  5520  5526
```

5532	5565	5578	5581	5587	5593	5599	5602	5605	5611	5617	5650	5693
5700	5703	5709	5715	5721	5724	5727	5733	5739	5772	5785	5788	5794
5800	5806	5809	5812	5818	5824	5857	5900	5907	5910	5916	5922	5928
5931	5934	5940	5946	5979	5982	5995	6001	6007	6013	6016	6019	6025
6031	6064	6107	6114	6117	6123	6129	6135	6138	6141	6147	6153	6186
6199	6202	6208	6214	6220	6223	6226	6232	6238	6271	6314	6321	6324
6330	6336	6342	6345	6348	6354	6360	6393	6406	6409	6415	6421	6427
6430	6433	6439	6445	6478	6521	6528	6531	6537	6543	6549	6552	6555
6561	6567	6600	6613	6616	6622	6628	6634	6637	6640	6646	6652	6685
6728	6735	6738	6744	6750	6756	6759	6762	6768	6774	6807	6820	6823
6829	6835	6841	6844	6847	6853	6859	6892	6935	6942	6945	6951	6957
6963	6966	6969	6975	6981	7014	7027	7030	7036	7042	7048	7051	7054
7060	7066	7099	7142	7149	7152	7158	7164	7170	7173	7176	7182	7188
7221	7234	7237	7243	7249	7255	7258	7261	7267	7273	7306	7347	7350
7353	7356	7359	7362	7368	7372	7374	7380	7386	7389	7392	7395	7398
7404	7419	7435	7438	7431	7437	7456	7462	7465	7521	7524	7527	7530
7533	7536	7539	7542	7545	7548	7554	7560	7564	7566	7572	7578	7581
7584	7587	7590	7593	7596	7599	7602	7611	7623	7626	7629	7632	7638
7656	7662	7665	7668	7696	7702	7705	7708	7764	7767	7770	7773	7776
7779	7782	7785	7788	7791	7794	7800	7806	7821	7830	7861	7907	7910
7913	7916	7919	7922	7928	7934	7946	7950	7952	7957	7970	7973	7976
7982	7997	8003	8006	8009	8021	8024	8027	8030	8045	8048	8063	8066
8085	8091	8094	8097	8142	8164	8170	8176	8182	8191	8194	8197	8207
8220	8226	8230	8235	8238	8254	8257	8299	8321	8327	8333	8339	8348
8351	8354	8364	8377	8383	8387	8395	8398	8414	8417	8459	8468	8471
8474	8483	8492	8495	8501	8504	8507	8530	8533	8542	8561	8564	8573
8582	8585	8591	8614	8617	8626	8672	8681	8684	8690	8693	8696	8702
8721	8724	8733	8742	8745	8751	8761	8774	8780	8783	8792	8838	8847
8850	8853	8862	8865	8868	8901	8910	8913	8916	8919	8970	8973	8976
8979	8982	8985	9095	9101	9116	9122	9128	9138	9151	9157	9166	9169
9187	9193	9199	9203	9222	9231	9233	9287	9290	9293	9296	9299	9302
9412	9433	9439	9445	9455	9468	9474	9483	9504	9510	9516	9526	9539
9548	9550	9604	9607	9610	9613	9616	9619	9729	9750	9756	9762	9772
9785	9791	9800	9821	9827	9833	9843	9856	9865	9867	9921	9924	9927
9930	9933	9936	10046	10067	10073	10079	10089	10102	10108	10117	10138	10144
10150	10160	10173	10182	10184	10238	10241	10244	10247	10250	10253	10363	10384
10390	10396	10406	10419	10455	10434	10455	10461	10467	10477	10490	10499	10501
10555	10558	10561	10564	10567	10570	10580	10701	10707	10713	10723	10736	10742
10751	10772	10778	10784	10794	10807	10816	10818	10872	10875	10878	10881	10884
10887	10997	11003	11018	11024	11030	11040	11053	11059	11068	11071	11089	11095
11101	11111	11124	11133	11135	11169	11192	11195	11198	11201	11204	11314	11335
11341	11347	11357	11370	11376	11385	11406	11412	11418	11428	11441	11450	11452
11506	11509	11512	11515	11518	11521	11631	11652	11658	11664	11674	11687	11693
11702	11723	11726	11735	11745	11750	11767	11769	11823	11826	11829	11832	11835
11838	11948	11969	11975	11981	11991	12004	12010	12019	12040	12046	12052	12062
12075	12084	12086	12140	12143	12146	12149	12152	12155	12265	12271	12286	12292
12298	12308	12321	12327	12336	12339	12357	12363	12369	12379	12392	12401	12403
12457	12460	12463	12466	12469	12472	12582	12588	12603	12609	12615	12633	12638
12644	12653	12656	12674	12680	12686	12696	12709	12718	12720	12774	12777	12780
12783	12786	12789	12899	12905	12920	12936	12952	12942	12965	12961	12970	12973
12991	12997	13003	13013	13026	13035	13037	13091	13094	13097	13100	13103	13106
13216	13237	13243	13249	13259	13272	13278	13287	13308	13314	13320	13330	13343
13352	13354	13408	13411	13414	13417	13420	13423	13533	13539	13554	13560	13566
13576	13589	13595	13604	13607	13625	13631	13637	13647	13660	13669	13671	13717
13720	13723	13726	13754	13789	13792	13795	13860	13863	13866	13876	13879	13882
13885	13888	13892	13895	13900	13903	13906	13909	13912	13915	13918	13921	13927

```
                    13930   13933   13936   13939   13942   13945   13948   13951   13954   13957   13962   13965   13968
                    13971   13980   13990   13996   14003   14009   14012   14037   14046   14204   14207   14217   14220
                    14223   14226   14236   14239   14249   14252   14262   14265   14275   14278   14288   14291   14301
                    14304   14314   14317   14327   14330   14349   14361   14370   14376   14389   14398   14417   14420
                    14426   14432   14435   14438   14444   14490   14493   14496   14499   14502   14505   14508   14514
                    14520   14535   14541   14553   14565   14607   14613   14626   14632   14638   14641   14644   14650
                    14702   14705   14708   14711   14720   14729   14753   14767   14770   14773   14809   14812   14815
                    14818   14827   14836   14842   14860   14863   14869   14872   14885   14888   14891   14897   14906
                    14912   14930   14933   14939   14942   14955   14961   14970   14976   14991   15013   15019   15025
.SCOP1  004360    1086#   1558
.SELA = 000200       1#   2337    2350    2356    2417    2430    2436    2534    2547    2553    2614    2627    2633
                    2731    2744    2750    2811    2824    2830    2928    2941    2947    3008    3021    3027    3135
                    3148    3154    3224    3237    3243    3361    3374    3380    3459    3472    3478    3576    3589
                    3595    3656    3669    3675    3773    3786    3792    3853    3866    3872    3970    3983    3989
                    4050    4063    4069    4167    4180    4186    4247    4260    4266    4364    4377    4383    4444
                    4457    4463    4561    4574    4580    4641    4654    4660    4758    4771    4777    4838    4851
                    4857    4955    4968    4974    5036    5048    5054    5165    5235    5249    5310    5374    5395
                    5496    5517    5581    5602    5703    5724    5788    5809    5910    5931    5995    6016    6117
                    6138    6202    6223    6324    6345    6409    6430    6531    6552    6616    6637    6738    6759
                    6823    6844    6945    6966    7030    7051    7152    7173    7237    7258    7389    7395    7428
                    7465    7581    7587    7593    7599    7623    7629    7665    7705    7967    7973    8045    8048
                    8063    8194    8207    8220    8226    8254    8257    8351    8364    8377    8383    8414    8417
                    8474    8492    8504    8582    8690    8742    8761    8774    8868    8919    9101    9138    9151
                    9157    9169    9209    9222    9455    9468    9474    9526    9539    9772    9785    9791    9843
                    9856   10089   10102   10108   10160   10173   10406   10419   10425   10477   10490   10723   10736
                   10742   10794   10907   11040   11053   11059   11111   11124   11357   11370   11376   11428   11441
                   11674   11687   11693   11745   11758   11991   12004   12010   12062   12075   12308   12321   12327
                   12379   12392   12625   12638   12644   12696   12709   12942   12955   12961   13013   13026   13259
                   13272   13278   13330   13343   13576   13589   13595   13647   13660   13795   13990   13996   14204
                   14217   14223   14236   14249   14262   14275   14288   14301   14314   14327   14420   14435   14456
                   14607   14626   14641   14860   14876   14930   14946   15001
.SELB = 000220       1#   2286    2294    2300    2312    2321    2327    2360    2374    2380    2392    2401    2407
                    2440    2463    2491    2497    2509    2518    2554    2567    2571    2577    2589    2598    2604
                    2637    2680    2688    2694    2706    2715    2721    2754    2768    2774    2786    2795    2801
                    2834    2877    2885    2891    2903    2912    2918    2951    2965    2971    2983    2992    2998
                    3031    3075    3083    3089    3096    3110    3119    3125    3158    3172    3178    3184    3199
                    3208    3214    3247    3253    3300    3306    3312    3321    3336    3345    3351    3384    3398
                    3404    3410    3419    3434    3443    3449    3462    3525    3533    3539    3551    3560    3566
                    3599    3613    3619    3631    3640    3646    3673    3722    3730    3736    3748    3757    3763
                    3796    3810    3816    3828    3837    3843    3876    3919    3927    3933    3945    3954    3960
                    3993    4007    4013    4035    4034    4040    4073    4116    4124    4130    4142    4151    4157
                    4190    4204    4210    4222    4231    4237    4270    4313    4321    4327    4339    4348    4354
                    4387    4401    4407    4419    4428    4434    4467    4510    4518    4524    4536    4545    4551
                    4584    4598    4604    4616    4625    4631    4664    4707    4715    4721    4733    4742    4748
                    4781    4795    4801    4813    4822    4828    4861    4904    4912    4918    4930    4939    4945
                    4978    4992    4998    5010    5019    5025    5058    5102    5109    5112    5115    5127    5130
                    5156    5179    5182    5185    5197    5200    5226    5279    5286    5292    5295    5307    5313
                    5319    5325    5335    5348    5354    5358    5371    5377    5380    5392    5398    5404    5410
                    5420    5433    5439    5443    5466    5493    5499    5502    5514    5520    5526    5532    5542
                    5555    5561    5565    5578    5584    5587    5599    5605    5611    5617    5627    5640    5646
                    5650    5693    5700    5706    5709    5721    5727    5733    5739    5749    5762    5768    5772
                    5785    5791    5794    5806    5812    5818    5824    5834    5847    5853    5857    5900    5907
                    5913    5916    5928    5934    5940    5946    5956    5969    5975    5979    5992    5998    6001
                    6013    6019    6025    6031    6041    6054    6060    6064    6107    6114    6120    6123    6135
                    6141    6147    6153    6163    6176    6182    6186    6199    6205    6208    6220    6226    6232
                    6238    6248    6261    6267    6271    6314    6321    6327    6330    6342    6348    6354    6360
```

```
                6370  6383  6389  6393  6406  6412  6415  6427  6433  6439  6445  6455  6468
                6474  6478  6521  6528  6534  6537  6549  6555  6561  6567  6577  6590  6596
                6600  6613  6619  6622  6634  6640  6646  6652  6662  6675  6681  6685  6728
                6735  6741  6744  6756  6762  6768  6774  6784  6797  6803  6807  6820  6826
                6829  6841  6847  6853  6859  6869  6882  6888  6892  6935  6942  6948  6951
                6963  6969  6975  6981  6991  7004  7010  7014  7027  7033  7036  7048  7054
                7060  7066  7076  7089  7095  7099  7142  7149  7155  7158  7170  7176  7182
                7188  7198  7211  7217  7221  7234  7240  7243  7255  7261  7267  7273  7283
                7296  7302  7306  7347  7350  7353  7356  7359  7362  7368  7372  7374  7380
                7386  7392  7398  7404  7425  7431  7437  7462  7471  7521  7524  7527  7530
                7533  7536  7539  7542  7545  7548  7560  7564  7566  7572  7578  7584  7590
                7596  7602  7626  7632  7638  7662  7668  7674  7702  7708  7714  7764  7767
                7770  7773  7776  7779  7782  7785  7788  7791  7794  7800  7806  7821  7842
                7907  7910  7913  7916  7919  7922  7928  7934  7946  7950  7970  7976  7982
                8021  8024  8027  8030  8042  8060  8066  8072  8142  8157  8160  8164  8167
                8170  8182  8185  8191  8197  8230  8299  8314  8317  8321  8324  8327  8339
                8342  8348  8354  8387  8459  8468  8471  8477  8489  8495  8501  8507  8561
                8564  8567  8579  8585  8591  8672  8681  8684  8721  8724  8727  8739  8745
                8751  8838  8847  8850  8853  8859  8878  8907  8970  8973  8976  8979  8985
                9098  9113  9116  9122  9128  9160  9172  9184  9187  9193  9199  9287  9290
                9293  9296  9302  9415  9430  9433  9439  9445  9477  9489  9501  9504  9510
                9516  9604  9607  9610  9613  9619  9732  9747  9750  9756  9762  9794  9806
                9818  9821  9827  9833  9921  9924  9927  9930  9936 10049 10064 10067 10073
               10079 10111 10123 10135 10138 10144 10150 10238 10241 10244 10247 10253 10366
               10381 10384 10390 10396 10428 10440 10452 10455 10461 10467 10555 10558 10561
               10564 10570 10683 10698 10701 10707 10713 10745 10757 10769 10772 10778 10784
               10872 10875 10878 10881 10887 11000 11015 11018 11024 11030 11062 11074 11086
               11089 11095 11101 11109 11192 11195 11198 11204 11317 11332 11335 11341 11347
               11379 11391 11403 11406 11412 11418 11506 11509 11512 11515 11521 11634 11649
               11652 11658 11664 11696 11708 11720 11723 11729 11735 11823 11826 11829 11832
               11838 11951 11966 11969 11975 11981 12013 12025 12037 12040 12046 12052 12140
               12143 12146 12149 12155 12246 12263 12266 12272 12275 12330 12342 12354 12357
               12363 12369 12457 12460 12463 12466 12472 12585 12600 12603 12609 12615 12647
               12659 12671 12674 12680 12686 12774 12777 12780 12783 12789 12902 12917 12920
               12926 12932 12964 12976 12980 13091 13097 13100 13106
               13219 13234 13237 13243 13249 13281 13293 13305 13308 13314 13320 13408 13411
               13414 13417 13423 13536 13551 13554 13560 13566 13588 13610 13622 13625 13631
               13637 13717 13720 13723 13726 13738 13754 13770 13808 13860 13863 13866 13875
               13879 13882 13885 13888 13892 13895 13900 13903 13906 13909 13912 13915 13918
               13921 13927 13930 13933 13936 13939 13942 13945 13948 13951 13954 13957 13962
               13971 14009 14046 14207 14220 14226 14239 14252 14265 14278 14291 14304 14317
               14330 14349 14355 14367 14370 14376 14417 14432 14438 14444 14490 14493 14496
               14499 14502 14505 14508 14514 14520 14526 14553 14565 14638 14644 14650 14702
               14705 14708 14711 14815 14818 14869 14872 14891 14897 14939 14942 14955 14961
               15013 15019 15025
.SIMM = 000000    18  2283  2290  2318  2324  2331  2344  2370  2398  2404  2411  2424  2447
               2480  2487  2515  2521  2528  2541  2567  2595  2601  2608  2621  2644  2677
               2684  2712  2718  2725  2738  2764  2792  2798  2805  2818  2841  2874  2881
               2909  2915  2922  2935  2961  2989  2995  3002  3015  3038  3072  3079  3086
               3116  3122  3129  3142  3168  3175  3205  3211  3218  3231  3254  3289  3296
               3303  3318  3342  3348  3355  3368  3394  3401  3416  3440  3446  3453  3466
               3489  3522  3529  3557  3563  3570  3583  3609  3637  3643  3650  3663  3686
               3719  3726  3754  3760  3767  3780  3806  3834  3840  3847  3860  3883  3916
               3923  3951  3957  3964  3977  4003  4031  4037  4044  4057  4080  4113  4120
               4148  4154  4161  4174  4200  4228  4234  4241  4254  4277  4310  4317  4345
               4351  4358  4371  4397  4425  4431  4438  4451  4474  4507  4514  4542  4548
```

```
4555  4568  4594  4622  4628  4635  4648  4671  4704  4711  4739  4745  4752
4765  4791  4819  4825  4832  4845  4868  4901  4908  4936  4942  4949  4962
4988  5016  5022  5029  5042  5065  5099  5106  5133  5137  5147  5176  5203
5207  5217  5243  5276  5283  5316  5422  5430  5442  5446  5401  5407  5414
5427  5450  5463  5490  5523  5529  5536  5549  5575  5608  5614  5621  5634
5657  5690  5697  5730  5736  5743  5756  5782  5815  5821  5828  5841  5864
5897  5904  5937  5943  5950  5963  5989  6022  6028  6035  6048  6071  6104
6111  6144  6150  6157  6170  6196  6229  6235  6242  6255  6278  6311  6318
6351  6357  6364  6377  6403  6436  6442  6449  6462  6485  6518  6525  6558
6564  6571  6584  6610  6643  6649  6656  6669  6692  6725  6732  6765  6771
6778  6791  6817  6850  6856  6863  6876  6890  6932  6939  6972  6978  6985
6998  7024  7057  7063  7070  7083  7106  7134  7146  7179  7185  7192  7205
7231  7264  7270  7277  7290  7313  7344  7365  7377  7383  7401  7434  7441
7468  7475  7485  7518  7557  7569  7575  7605  7635  7671  7678  771i  7718
7728  7761  7797  7803  7809  7818  7839  7846  7855  7865  7898  7901  7904
7925  7931  7937  7940  7943  7961  7984  7979  8015  8018  8033  8036  8039
8069  8076  8104  8139  8145  8148  8151  8154  8188  8201  8214  8241  8264
8296  8302  8305  8308  8311  8345  8358  8371  8401  8424  8456  8463  8465
8498  8511  8521  8536  8552  8555  8558  8588  8595  8605  8620  8637  8669
8675  8678  8687  8712  8715  8718  8748  8755  8768  8786  8803  8835  8841
8844  8856  8875  8882  8892  8904  8926  8961  8984  8987  8988  8991  8994
8997  9000  9003  9006  9009  9012  9015  9018  9021  9024  9027  9030  9033
9036  9039  9042  9045  9048  9051  9054  9057  9060  9063  9066  9069  9072
9075  9078  9081  9084  9088  9092  9119  9125  9132  9145  9163  9190  9196
9203  9216  9228  9243  9278  9281  9284  9305  9308  9311  9314  9317  9320
9323  9326  9329  9332  9335  9338  9341  9344  9347  9350  9353  9356  9359
9362  9365  9368  9371  9374  9377  9380  9383  9386  9389  9392  9395  9398
9401  9405  9409  9436  9442  9449  9462  9480  9507  9513  9520  9533  9545
9560  9595  9598  9601  9622  9625  9628  9631  9634  9637  9640  9643  9646
9649  9652  9655  9658  9661  9664  9667  9670  9673  9676  9679  9682  9685
9688  9691  9694  9697  9700  9703  9706  9709  9712  9715  9718  9722  9726
9753  9759  9766  9779  9797  9824  9830  9837  9850  9862  9877  9912  9915
9918  9939  9942  9945  9948  9951  9954  9957  9980  9983  9986  9989  9972
9975  9978  9981  9984  9987  9990  9993  9996  9999  10002  10005  10008  10011
10014  10017  10020  10023  10026  10029  10032  10035  10039  10043  10070  10076  10083
10096  10114  10141  10147  10154  10167  10176  10194  10229  10232  10235  10256  10259
10262  10265  10268  10271  10274  10277  10280  10283  10286  10289  10292  10295  10298
10301  10304  10307  10310  10313  10316  10319  10322  10325  10328  10331  10334  10337
10340  10343  10346  10349  10362  10356  10540  10387  10393  10400  10413  10431  10458
10464  10471  10484  10496  10511  10546  10549  10552  10573  10576  10579  10582  10585
10588  10591  10594  10597  10600  10603  10606  10609  10612  10615  10618  10621  10624
10627  10630  10633  10636  10639  10642  10645  10648  10651  10654  10657  10660  10663
10666  10669  10673  10677  10794  10710  10717  10730  10738  10775  10781  10788  10801
10813  10828  10863  10866  10869  10840  10843  10896  10898  10902  10905  10908  10911
10914  10917  10920  10923  10926  10929  10932  10935  10938  10941  10944  10947  10950
10953  10956  10959  10962  10965  10968  10971  10974  10977  10980  10983  10986  10990
10994  11021  11027  11034  11047  11045  11092  11098  11105  11118  11130  11145  11180
11183  11186  11207  11210  11213  11216  11219  11222  11225  11228  11231  11234  11237
11240  11243  11246  11249  11252  11255  11258  11261  11264  11267  11270  11273  11276
11279  11282  11285  11288  11291  11294  11297  11300  11303  11307  11311  11338  11344
11351  11364  11382  11409  11415  11422  11435  11447  11462  11497  11500  11503  11524
11527  11530  11533  11536  11539  11542  11545  11548  11551  11554  11557  11560  11563
11566  11569  11572  11575  11578  11581  11584  11587  11590  11593  11596  11599  11602
11605  11608  11611  11614  11617  11620  11624  11628  11655  11661  11668  11681  11699
11726  11732  11739  11752  11764  11779  11814  11817  11820  11841  11844  11847  11850
11853  11856  11859  11862  11865  11868  11871  11874  11877  11880  11883  11886  11889
```

```
                11892  11895  11898  11901  11904  11907  11910  11913  11916  11919  11922  11925  11928
                11931  11934  11937  11941  11945  11972  11978  11985  11998  12016  12043  12049  12056
                12069  12081  12096  12131  12134  12137  12158  12161  12164  12167  12170  12173  12176
                12179  12182  12185  12188  12191  12194  12197  12200  12203  12206  12209  12212  12215
                12218  12221  12224  12227  12230  12233  12236  12239  12242  12245  12248  12251  12254
                12258  12262  12269  12295  12302  12315  12333  12360  12365  12373  12386  12398  12413
                12448  12451  12454  12475  12478  12481  12484  12487  12490  12493  12496  12499  12502
                12505  12508  12511  12514  12517  12520  12523  12526  12529  12532  12535  12538  12541
                12544  12547  12550  12553  12556  12559  12562  12565  12568  12571  12575  12579  12606
                12612  12619  12632  12650  12677  12683  12690  12703  12715  12730  12765  12768  12771
                12782  12785  12788  12801  12804  12807  12810  12813  12816  12819  12822  12825  12828
                12831  12834  12837  12840  12843  12846  12849  12852  12855  12858  12861  12864  12867
                12870  12873  12876  12879  12882  12885  12888  12892  12895  12923  12929  12936  12949
                12967  12994  13000  13007  13020  13032  13047  13062  13085  13088  13104  13112  13115
                13118  13121  13124  13127  13130  13133  13136  13139  13142  13145  13148  13151  13154
                13157  13160  13163  13166  13169  13172  13175  13178  13181  13184  13187  13190  13193
                13196  13199  13202  13205  13209  13213  13240  13246  13253  13266  13284  13311  13317
                13324  13337  13349  13364  13399  13402  13405  13428  13429  13432  13435  13438  13441
                13444  13447  13450  13453  13456  13459  13462  13465  13468  13471  13474  13477  13480
                13483  13486  13489  13492  13495  13498  13501  13504  13507  13510  13513  13516  13519
                13522  13526  13530  13557  13563  13570  13583  13601  13628  13634  13641  13654  13666
                13681  13714  13736  13742  13751  13767  13774  13805  13812  13822  13857  13870  13873
                13974  14021  14027  14031  14034  14053  14063  14073  14083  14093  14103  14113  14125
                14135  14145  14155  14166  14175  14185  14195  14211  14230  14243  14256  14269  14282
                14295  14308  14321  14334  14343  14346  14362  14373  14380  14408  14441  14448  14481
                14484  14487  14511  14517  14523  14529  14538  14550  14562  14647  14654  14664  14699
                14714  14723  14747  14821  14833  14839  14866  14894  14903  14909  14936  14958  14967
                14973  14985  15010  15016  15022  15041
.SIN0 = 020000    1#  3536  3554  3616  3634  3733  3751  3813  3831  3930  3948  4010  4028
                4127  4145  4207  4225  4324  4342  4404  4422  4521  4539  4601  4619  4718
                4736  4798  4816  4915  4933  4995  5013  7413  7450  7647  7650  7687  7690
                7991  7994  14121  14131  14141  14151  14161  14171  14181  14191  14604  14623
.SIN1 = 120000    1#  2247  2315  2377  2395  2494  2512  2574  2592  2691  2709  2771  2789
                2888  2906  2968  2986  3098  3113  3187  3205  3315  3333  3413  3437  7407
                7551  7608  7614  7641  7812  7824  7850  7905  13729  13757  13783  14049  14059
                14069  14079  14089  14099  14109  14532  14544  14556  14568  14574  14580  14583  14589
                14595  14598  14717  14726  14732  14738  14750  14758  14784  14824  14830  14845  14851
                14900  14915  14921  14964  14979  14988  14994  15028  15034  15047
.SMEM = 040000    1#  2321  2327  2401  2407  2518  2524  2598  2604  2715  2721  2795  2801
                2912  2918  2992  2998  3089  3119  3125  3178  3208  3214  3306  3321  3345
                3351  3404  3419  3443  3449  3560  3566  3640  3646  3757  3763  3837  3843
                3954  3960  4034  4040  4151  4157  4231  4237  4348  4354  4428  4434  4545
                4551  4625  4631  4742  4748  4822  4828  4939  4945  5019  5025  5109  5112
                5121  5127  5156  5179  5185  5191  5197  5326  5332  5336  5348  5354  5377
                5420  5433  5439  5499  5505  5555  5561  5650  5657  5660  5676  5706  5749
                5762  5768  5791  5834  5847  5853  5913  5956  5969  5975  5998  6041  6054
                6060  6120  6163  6176  6182  6205  6248  6261  6267  6327  6370  6383  6389
                6412  6455  6468  6474  6534  6577  6590  6596  6619  6662  6675  6681  6741
                6784  6797  6803  6826  6869  6882  6888  6948  6991  7004  7010  7033  7076
                7089  7095  7155  7198  7211  7217  7240  7283  7296  7302  7471  7674  7714
                7815  7842  8042  8054  8060  8072  8157  8160  8167  8185  8247  8314  8317
                8324  8342  8407  8477  8489  8567  8579  8727  8739  8859  8878  8907  9098
                9107  9113  9160  9172  9178  9184  9415  9418  9424  9430  9477  9486  9489
                9496  9501  9732  9736  9741  9747  9794  9803  9806  9812  9818  10049  10052
                10058  10064  10111  10120  10123  10129  10135  10366  10369  10375  10381  10428  10437
                10440  10446  10452  10683  10686  10692  10698  10745  10754  10757  10763  10769  11000
```

```
                    11009  11015  11062  11074  11080  11086  11317  11320  11326  11332  11379  11388  11391
                    11397  11403  11634  11637  11643  11649  11696  11705  11708  11714  11720  11951  11954
                    11960  11966  12013  12022  12025  12031  12037  12268  12277  12283  12330  12342  12348
                    12354  12586  12594  12600  12647  12659  12665  12671  12902  12911  12917  12964  12976
                    12982  12988  13219  13222  13228  13234  13281  13290  13293  13299  13305  13536  13545
                    13551  13598  13610  13616  13622  13738  13770  13808  14355  14367  14526
.START  002402        233    732#   748   1828
.SUBMC= 000040          1#  11954  12022
.SUB1C= 000340          1#   6542   6626   8702   8792  11320  11388
.SUB2C= 000360            1#  2306   2386   2503   2583   2700   2780   2897   2977   3104   3193   3330   3428
                     3545   3625   3742   3822   3939   4019   4136   4216   4333   4413   4530   4610   4727
                     4807   4924   5004   5121   5191   5301   5386   5508   5593   5715   5800   5922   6007
                     6129   6214   6336   6421   6543   6628   6750   6835   6957   7042   7164   7249   7419
                     7456   7656   7696   8054   8176   8247   8333   8407   8483   8573   8733   9107   9178
                     9424   9496   9741   9812  10058  10129  10375  10446  10692  10763  11009  11080  11326
                    11397  11643  11714  11960  12031  12277  12348  12594  12665  12911  12982  13222  13228
                    13290  13299  13545  13616  13980  14361  14426  14613  14632
.SO    = 020000        1#
.TIMER  007464      1565   1752#
.XOR   = 000320        1#  10369  10437
.$ASTA= ***** U     1178   1181
.SX    = 002034     486#   491
.S.    = 104510    2283#  2284   2286#  2287   2290#  2291   2294#  2295   2297#  2298   2300#  2301   2306#
                    2307   2309#  2310#  2312#  2313   2315#  2316   2318#  2319   2321#  2322   2324#  2325
                    2327#  2328   2331#  2332   2334#  2336#  2337#  2338   2340#  2341#  2344#  2345   2347#
                    2348#  2350#  2351   2353#  2354#  2356#  2357   2362#  2363#  2365#  2366#  2370#  2371
                    2374#  2375   2377#  2378   2380#  2381   2386#  2387   2389#  2390#  2392#  2393   2395#
                    2396   2398#  2399   2401#  2402   2404#  2405   2407#  2408   2411#  2412   2414#  2415#
                    2417#  2418   2420#  2421#  2424#  2426   2427#  2428#  2430#  2431   2433#  2434#  2436#
                    2437   2442#  2443#  2447#  2448   2450#  2451#  2453#  2454#  2480#  2481   2483#  2484
                    2487#  2488   2491#  2492   2494#  2496   2497#  2498   2503#  2504   2506#  2507#  2509#
                    2510   2512#  2513   2534#  2516   2517#  2519   2521#  2542   2544#  2525   2528#  2529
                    2531#  2532#  2534#  2536#  2538#  2539#  2541#  2542   2543#  2545#  2547#  2548   2550#
                    2551#  2553#  2554#  2556#  2558#  2559#  2562#  2563#  2567#  2571#  2572   2574#  2575
                    2577#  2578   2583#  2584#  2586#  2587#  2589#  2590   2591#  2593   2595#  2596   2598#
                    2599   2601#  2602   2604#  2605   2607#  2608#  2609   2611#  2614#  2615   2617#  2618#
                    2621#  2622   2624#  2625#  2627#  2629#  2630#  2631#  2633#  2634   2639#  2640#  2644#
                    2645   2647#  2648#  2650#  2651#  2677#  2678   2679#  2681   2684#  2685   2688#  2689
                    2691#  2692   2694#  2695   2700#  2701   2702#  2703#  2706#  2707   2709#  2710   2712#
                    2713   2715#  2716   2718#  2719   2721#  2722   2725#  2726   2728#  2729#  2731#  2732
                    2734#  2735#  2738#  2739   2741#  2742#  2744#  2745   2747#  2748#  2750#  2751   2756#
                    2757#  2759#  2760#  2764#  2765   2768#  2769   2771#  2772   2774#  2775   2780#  2781
                    2783#  2784#  2786#  2787   2789#  2790   2792#  2793   2795#  2796   2798#  2799   2801#
                    2802   2805#  2806   2808#  2809#  2811#  2812   2814#  2815#  2818#  2819   2821#  2822#
                    2824#  2825   2827#  2828#  2830#  2831   2836#  2837#  2841#  2842   2844#  2845#  2847#
                    2848#  2874#  2875   2877#  2878   2881#  2882   2885#  2886   2888#  2889   2891#  2892
                    2897#  2898   2900#  2901#  2903#  2904   2906#  2907   2909#  2910   2912#  2913   2915#
                    2916   2918#  2919   2922#  2923   2925#  2926#  2928#  2929   2931#  2932#  2935#  2936
                    2938#  2939#  2941#  2943   2944#  2945#  2947#  2948   2953#  2954#  2956#  2957#  2961#
                    2962   2965#  2966   2968#  2969   2971#  2972   2977#  2978   2980#  2981#  2983#  2984
                    2986#  2987   2989#  2990   2992#  2993   2995#  2996   2998#  2999   3002#  3003   3005#
                    3006#  3008#  3009   3011#  3012#  3015#  3016   3018#  3019#  3021#  3022   3024#  3025#
                    3027#  3028   3033#  3034#  3038#  3039   3041#  3042#  3044#  3045#  3072#  3073   3075#
                    3076   3079#  3080   3083#  3084   3086#  3087   3089#  3090   3092#  3093   3095#  3096
                    3098#  3099   3104#  3105   3107#  3108#  3110#  3111   3113#  3114   3116#  3117   3119#
                    3120   3122#  3123   3125#  3126   3129#  3130   3132#  3133#  3135#  3136   3138#  3139#
```

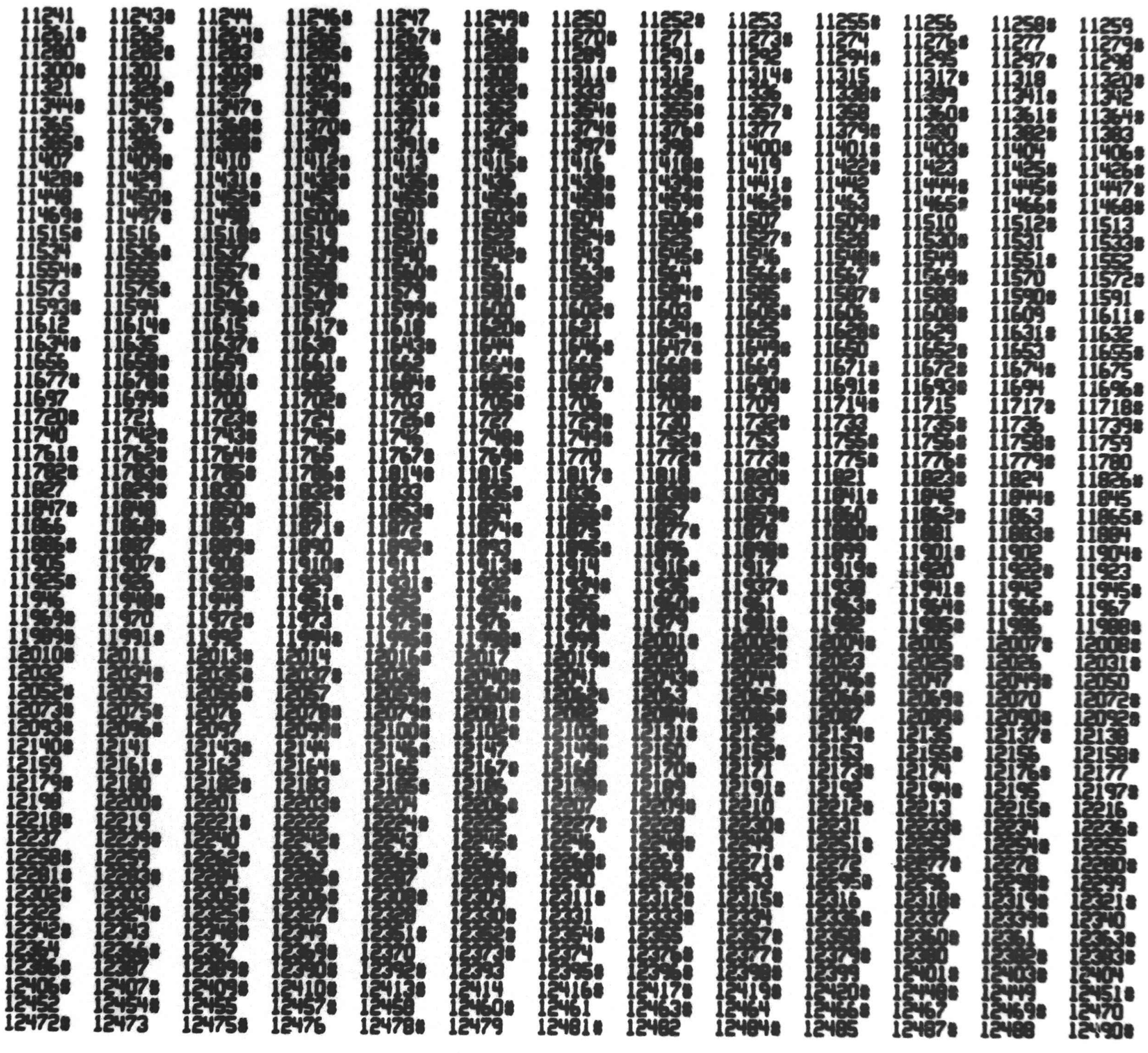
3142# 3143 3145# 3146# 3148# 3149 3151# 3152# 3154# 3155 3160# 3161# 3163#
3164# 3168# 3169 3172# 3173 3175# 3176 3178# 3179 3181# 3182 3184# 3185
3187# 3188 3193# 3194 3196# 3197# 3199# 3200 3202# 3203 3205# 3206 3208#
3209 3211# 3212 3214# 3215 3218# 3219 3221# 3222# 3224# 3225 3227# 3228#
3231# 3232 3244# 3245# 3247# 3248# 3240# 3241# 3243 3244 3249# 3250# 3254#
3255 3257# 3258# 3260# 3261# 3262# 3262 3242# 3243 3246# 3246 3247 3300# 3301
3303# 3304 3306# 3307 3309# 3310 3312# 3313 3315# 3316 3318# 3319 3321#
3322 3324# 3325 3330# 3331 3333# 3334# 3336# 3337 3339# 3340 3342# 3343
3345# 3346 3348# 3349 3351# 3353 3354# 3355# 3358# 3359# 3360# 3361# 3362 3364#
3365# 3368# 3369 3371# 3372# 3374# 3375 3377# 3378# 3380# 3381 3386# 3387#
3389# 3390# 3394# 3395 3398# 3399 3401# 3402 3404# 3405 3407# 3408 3410#
3411 3413# 3414 3416# 3417 3419# 3420 3422# 3423 3426# 3428# 3429 3431# 3432#
3434# 3436 3437# 3438 3440# 3441 3443# 3444 3446# 3447 3449# 3450 3453#
3454 3456# 3457# 3458 3460 3462# 3463# 3466# 3467 3469# 3470# 3472# 3473
3475# 3476# 3478# 3479 3481# 3483# 3484# 3485# 3487# 3488# 3492# 3495# 3496# 3522#
3523 3525# 3526 3527# 3530 3532 3534# 3535# 3537 3538# 3540 3545# 3546
3548# 3549# 3551# 3552 3554# 3555# 3557# 3558# 3560# 3561 3563# 3564 3566#
3567 3570# 3571 3573# 3574# 3576# 3577 3579# 3580# 3583# 3584# 3586# 3587#
3589# 3590 3592# 3593# 3595# 3596# 3601# 3602# 3604# 3605# 3609# 3610 3613#
3614 3616# 3617 3619# 3620 3625# 3626 3628# 3629# 3631# 3632 3634# 3635
3637# 3638 3640# 3641 3643# 3644 3646# 3647 3650# 3651 3653# 3654# 3656#
3657 3659# 3660# 3663# 3664 3666# 3667# 3669# 3670 3672# 3673# 3675# 3676
3681# 3682# 3686# 3687 3689# 3690# 3692# 3693# 3719# 3720 3722# 3723 3726#
3727 3730# 3731 3733# 3734 3736# 3737 3742# 3743 3745# 3746# 3748# 3749
3751# 3752 3754# 3755 3757# 3758 3760# 3761 3763# 3764 3767# 3768 3770#
3771# 3773# 3774 3776# 3777# 3780# 3781 3783# 3784# 3786# 3787 3789# 3790#
3792# 3793 3798# 3799# 3801# 3802# 3806# 3807 3810# 3811 3813# 3814 3816#
3817 3822# 3823 3825# 3826# 3828# 3829 3831# 3832 3834# 3835 3837# 3838
3840# 3841 3843# 3844 3846# 3848 3850# 3851# 3853# 3854 3856# 3857# 3860#
3861 3863# 3864# 3866# 3867 3869# 3870 3872# 3873 3876# 3878# 3879# 3883# 3884
3886# 3887# 3888# 3890# 3891# 3916# 3917 3919# 3923# 3924 3926# 3927# 3928 3930#
3931 3933# 3934 3936# 3940 3941# 3942# 3943# 3945# 3946# 3948# 3949 3951# 3952
3954# 3955 3957# 3958 3960# 3961# 3963# 3964 3966# 3967# 3968# 3969# 3970# 3971 3973#
3974# 3977# 3978 3980# 3981# 3983# 3984 3986# 3987# 3988# 3989# 3990 3995# 3996#
3998# 3999# 4003# 4004 4007# 4008 4010# 4011 4013# 4014 4019# 4020 4022#
4023# 4025# 4026 4028# 4029 4031# 4032 4034# 4035 4037# 4038 4040# 4041
4044# 4045 4047# 4048# 4050# 4051 4053# 4054# 4057# 4058 4060# 4061# 4063#
4064 4066# 4067# 4069# 4070 4075# 4076# 4080# 4081 4083# 4084 4086# 4087#
4113# 4114 4116# 4117 4120# 4121 4124# 4125 4127# 4128 4130# 4131 4136#
4137 4139# 4140# 4142# 4143 4145# 4146 4148# 4149 4151# 4152 4154# 4155
4157# 4158 4161# 4162 4164# 4165# 4167# 4168 4170# 4171# 4174# 4175 4177#
4178# 4180# 4181 4183# 4184# 4186# 4187 4192# 4193# 4195# 4196# 4200# 4201
4204# 4205 4207# 4208 4210# 4211 4216# 4217 4219# 4220# 4222# 4223 4225#
4226 4228# 4229 4231# 4232 4234# 4235 4237# 4238 4241# 4242 4244# 4245#
4247# 4248 4250# 4251# 4254# 4255 4257# 4258# 4260# 4261 4263# 4264# 4266#
4267 4272# 4273# 4277# 4278 4280# 4281# 4283# 4284# 4310# 4311 4313# 4314
4317# 4318 4321# 4322 4324# 4325 4327# 4328 4333# 4334 4336# 4337# 4339#
4340 4342# 4343 4345# 4346 4348# 4349 4351# 4352 4354# 4355 4358# 4359
4361# 4362# 4364# 4365 4367# 4368# 4371# 4372 4374# 4375# 4377# 4378 4380#
4381# 4383# 4384 4389# 4390# 4392# 4393# 4397 4398 4401# 4402 4404# 4405
4407# 4408 4413# 4414 4416# 4417# 4419# 4421# 4422# 4423 4425# 4426 4428#
4429 4431# 4432 4434# 4435 4438# 4439 4441# 4442# 4444# 4445 4447# 4448#
4451# 4452 4454# 4455# 4457# 4458 4460# 4461# 4463# 4464 4469# 4470# 4474#
4475 4477# 4478# 4480# 4481# 4507# 4508 4510# 4511 4514# 4515 4518# 4519
4521# 4522 4524# 4525 4530# 4531 4533# 4534# 4536# 4537 4539# 4540 4542#
4543 4545# 4546 4548# 4549 4551# 4552 4555# 4556 4558# 4559# 4561# 4562

```
4564#  4565#  4568#  4569   4571#  4572#  4574#  4575   4577#  4578#  4580#  4581   4586#
4587#  4589#  4590#  4594#  4595   4598#  4599   4601#  4602   4604#  4605   4610#  4611
4613#  4614#  4616#  4617   4619#  4620   4622#  4623   4625#  4626   4628#  4629   4631#
4632   4634#  4636   4638#  4639#  4641#  4642   4644#  4645#  4648#  4649   4651#  4652#
4654#  4655   4657#  4658#  4660#  4621   4666#  4667#  4671#  4672   4674#  4675#  4677#
4678#  4704#  4705   4707#  4708   4711#  4712   4715#  4716   4718#  4719   4721#  4722
4727#  4728   4730#  4731#  4733#  4734   4736#  4737   4739#  4740   4742#  4743   4745#
4746   4748#  4749   4752#  4753   4755#  4756#  4758#  4759   4761#  4762#  4765#  4766
4768#  4769#  4771#  4772   4774#  4775#  4777#  4778   4783#  4784#  4786#  4787#  4791#
4792   4795#  4796   4798#  4799   4801#  4802   4807#  4808   4810#  4811#  4813#  4814
4816#  4817   4819#  4820   4822#  4823   4825#  4826   4828#  4829   4832#  4833   4835#
4836#  4838#  4839   4841#  4842#  4845#  4846   4848#  4849#  4851#  4852   4854#  4855#
4857#  4858   4863#  4864#  4868#  4863   4871#  4872#  4874#  4875#  4901#  4902   4904#
4905   4908#  4909   4912#  4913   4915#  4916   4918#  4919   4924#  4925   4927#  4928#
4930#  4931   4933#  4934   4936#  4937   4939#  4940   4942#  4943   4945#  4946   4949#
4950   4952#  4953#  4955#  4956#  4958#  4959#  4962#  4963   4965#  4966#  4968#  4969
4971#  4972#  4974#  4975   4980#  4981#  4983#  4984#  4988#  4989   4992#  4993   4995#
4996   4998#  4999   5004#  5005   5007#  5008#  5010#  5011   5013#  5014   5016#  5017
5019#  5020   5022#  5023   5025#  5026   5028#  5030   5032#  5033#  5035#  5036   5038#
5039#  5042#  5043   5045#  5046#  5048#  5049   5051#  5052#  5054#  5055   5060#  5061#
5065#  5066   5068#  5069#  5071#  5072#  5099#  5100   5102#  5103   5106#  5107   5109#
5110   5112#  5113   5115#  5116   5121#  5122   5124#  5125#  5127#  5128   5130#  5131
5133#  5134   5137#  5138#  5140#  5141#  5143#  5144#  5147#  5148   5150#  5151#  5153#
5154#  5156#  5157   5159#  5160   5162#  5163   5165#  5166   5168#  5169#  5171#  5172#
5176#  5177   5179#  5180   5182#  5183#  5185#  5186   5191#  5192   5194#  5195#  5197#
5198   5200#  5201   5203#  5204   5207#  5208   5210#  5211#  5213#  5214#  5217#  5218
5220#  5221#  5223#  5224#  5226#  5227   5229#  5230   5232#  5233   5235#  5236   5238#
5239#  5243#  5244   5246#  5247#  5249#  5250#  5276#  5277   5279#  5280   5283#  5284#
5286#  5287   5289#  5290   5292#  5293   5295#  5296   5301#  5302   5304#  5305#  5307#
5308   5310#  5311   5313#  5314   5316#  5317   5319#  5320   5322#  5323   5325#  5326
5329#  5330   5332#  5333#  5334#  5336#  5340#  5341#  5342#  5343   5345#  5346#  5348#
5349   5351#  5352#  5353#  5360#  5361#  5365#  5366#  5368#  5369   5371#  5372
5374#  5375   5377#  5378#  5380#  5381   5384#  5385#  5387#  5388   5392#  5393   5395#
5396   5398#  5399   5401#  5402   5404#  5405   5407#  5408   5410#  5411   5414#  5415
5417#  5418#  5420#  5421   5423#  5424#  5427#  5428#  5430#  5431#  5433#  5434   5436#
5437#  5439#  5440   5442#  5443#  5450#  5451   5453#  5454#  5456#  5457#  5463#  5464
5486#  5487   5490#  5491   5493#  5494#  5496#  5497#  5498#  5500   5502#  5503   5508#
5509   5511#  5512#  5514#  5515   5517#  5518#  5520#  5521   5523#  5524   5526#  5527
5529#  5530   5532#  5533   5536#  5537   5539#  5540#  5542#  5543   5545#  5546#  5549#
5550   5552#  5553#  5555#  5556#  5558#  5559#  5561#  5562   5567#  5568   5570#  5571#
5575#  5576   5578#  5579#  5581#  5582#  5584#  5585#  5587#  5588   5590#  5594   5596#
5597#  5599#  5600   5602#  5603   5605#  5606#  5608#  5609#  5611#  5612   5614#  5615
5617#  5618   5621#  5622   5624#  5625#  5627#  5628#  5630#  5631#  5634#  5635   5637#
5638#  5640#  5641   5643#  5644#  5646#  5647   5649#  5650#  5657#  5658   5660#  5661#
5643#  5644#  5690#  5691   5693#  5694#  5696#  5697#  5698#  5700#  5703#  5704   5706#
5707   5709#  5710   5715#  5716   5718#  5719#  5721#  5722   5724#  5726   5727#  5728
5730#  5731   5733#  5734   5736#  5737   5739#  5740   5743#  5744   5746#  5747#  5749#
5750   5752#  5753#  5756#  5757   5759#  5760#  5762#  5763   5765#  5766#  5768#  5769
5774#  5775#  5777#  5778#  5782#  5783   5785#  5786   5788#  5789   5791#  5792   5794#
5795   5800#  5801   5803#  5804#  5806#  5807   5809#  5810   5812#  5813   5815#  5816
5818#  5819   5821#  5822   5824#  5825   5828#  5829   5831#  5832#  5834#  5835   5837#
5838#  5841#  5842   5844#  5845#  5847#  5848   5850#  5851#  5853#  5854   5859#  5860#
5864#  5865   5867#  5868#  5870#  5871#  5897#  5898   5900#  5901   5904#  5905   5907#
5908   5910#  5911   5913#  5914   5916#  5917   5922#  5923   5925#  5926#  5928#  5929
5931#  5932   5934#  5935   5937#  5938   5940#  5941   5943#  5944   5946#  5947   5950#
5951   5953#  5954#  5956#  5957   5959#  5960#  5963#  5964   5966#  5967#  5969#  5970
```

```
5972#  5973#  5975#  5976   5981#  5982#  5984#  5985#  5989#  5990   5992#  5993   5995#
5996   5998#  5999   6001#  6002   6007#  6008   6010#  6011#  6013#  6014   6016#  6017
6019#  6020   6022#  6023   6025#  6026   6028#  6029   6031#  6032   6035#  6036   6038#
6039#  6041#  6042   6044#  6045#  6048#  6049   6051#  6053#  6054#  6055   6057#  6058#
6060#  6061   6066#  6067#  6071#  6072   6074#  6075#  6077#  6078#  6104#  6105   6107#
6108   6111#  6112   6114#  6115   6117#  6118   6120#  6121   6123#  6124   6129#  6130
6132#  6133#  6135#  6136   6138#  6139   6141#  6142   6144#  6145   6147#  6148   6150#
6151   6153#  6154   6157#  6158   6160#  6161#  6163#  6164   6166#  6167#  6170#  6171
6173#  6174#  6175#  6177   6179#  6180#  6182#  6183   6188#  6189#  6191#  6192#  6196#
6197   6199#  6200   6202#  6203   6205#  6206   6208#  6209   6214#  6215   6217#  6218#
6220#  6221   6223#  6224   6225#  6227   6229#  6230   6232#  6233   6235#  6236   6238#
6239   6242#  6243   6245#  6246#  6248#  6249   6251#  6252#  6255#  6256   6258#  6259#
6261#  6262   6264#  6265#  6267#  6268   6271#  6273#  6274#  6278#  6279   6281#  6282#  6284#
6285#  6311#  6312   6314#  6315   6318#  6319   6321#  6322   6324#  6325   6327#  6328
6330#  6331   6336#  6337   6339#  6340#  6342#  6343   6345#  6346   6348#  6349   6351#
6352   6354#  6356   6357#  6358   6360#  6361   6364#  6365   6367#  6368#  6370#  6371
6373#  6374#  6377#  6378   6380#  6381#  6383#  6384   6386#  6387#  6389#  6390   6395#
6396#  6398#  6399#  6403#  6404   6406#  6407   6409#  6410   6412#  6413   6415#  6416
6421#  6422   6424#  6425#  6427#  6428   6430#  6431   6433#  6434   6436#  6437   6439#
6440   6442#  6443   6445#  6446   6449#  6450   6452#  6453#  6455#  6456   6458#  6459#
6462#  6463   6465#  6466#  6468#  6469   6471#  6472#  6474#  6475   6480#  6481#  6485#
6486   6488#  6489#  6491#  6492#  6518#  6519   6521#  6522   6525#  6526   6528#  6529
6531#  6532   6534#  6536   6537#  6538   6543#  6544   6546#  6547#  6549#  6550   6552#
6553   6555#  6556   6558#  6559   6561#  6562   6564#  6565   6567#  6568   6571#  6572
6574#  6575#  6577#  6578   6580#  6581#  6584#  6585   6587#  6588#  6590#  6591   6593#
6594#  6596#  6597   6602#  6603#  6605#  6606#  6610#  6611   6613#  6614   6616#  6617
6619#  6620   6622#  6623   6628#  6629   6631#  6632#  6634#  6635   6637#  6638   6640#
6641   6643#  6644   6646#  6647   6649#  6650   6652#  6653   6656#  6657   6659#  6660#
6662#  6663   6665#  6666#  6669#  6670   6672#  6673#  6675#  6676   6678#  6679#  6681#
6682   6687#  6688#  6692#  6693   6695#  6696#  6698#  6699#  6725#  6726   6728#  6729
6732#  6733   6735#  6736   6738#  6739   6741#  6742   6744#  6745   6750#  6751   6753#
6754#  6756#  6757   6759#  6760   6762#  6763   6765#  6766   6768#  6769   6771#  6772
6774#  6775   6778#  6779   6781#  6782#  6784#  6785   6787#  6788#  6791#  6792   6794#
6795#  6797#  6798   6800#  6801#  6803#  6804   6809#  6810#  6812#  6813#  6817#  6818
6820#  6821   6823#  6824   6826#  6827   6829#  6830   6835#  6836   6838#  6839#  6841#
6842   6844#  6845   6847#  6848   6850#  6851   6853#  6854   6856#  6857   6859#  6860
6863#  6864   6866#  6867#  6869#  6870   6872#  6873#  6876#  6877   6879#  6880#  6882#
6883   6885#  6886#  6888#  6889   6891#  6892   6894#  6900   6902#  6903#  6905#  6906#
6932#  6933   6935#  6936   6938#  6940   6942#  6943   6945#  6946   6948#  6949   6951#
6952   6957#  6958   6960#  6961#  6963#  6964   6966#  6967   6969#  6970   6972#  6973
6975#  6976   6978#  6979   6981#  6982   6985#  6986   6988#  6989#  6991#  6992   6994#
6995#  6998#  6999   7001#  7002#  7004#  7005   7007#  7008#  7010#  7011   7016#  7017#
7019#  7020#  7024#  7025   7027#  7028   7030#  7031   7033#  7034   7036#  7037   7042#
7043   7045#  7046#  7048#  7049   7051#  7052   7054#  7055   7057#  7058   7060#  7061
7063#  7064   7066#  7067   7070#  7071   7073#  7074#  7076#  7077   7079#  7080#  7083#
7084   7086#  7087#  7089#  7090   7092#  7093#  7095#  7096   7101#  7102#  7106#  7107
7109#  7110#  7112#  7113#  7139#  7140   7142#  7143   7146#  7147   7149#  7150   7152#
7153   7155#  7156   7158#  7159   7164#  7165   7167#  7168#  7170#  7171   7173#  7174
7176#  7177   7179#  7180   7182#  7183   7185#  7186   7188#  7189   7192#  7193   7195#
7196#  7198#  7199   7201#  7202#  7205#  7206   7208#  7209#  7211#  7212   7214#  7215#
7217#  7218   7223#  7224#  7226#  7227#  7231#  7232   7234#  7235   7237#  7238   7240#
7241   7243#  7244   7249#  7250   7252#  7253#  7255#  7256   7258#  7263   7261#  7262
7264#  7265   7267#  7268   7270#  7271   7273#  7274   7277#  7278   7280#  7281#  7283#
7284   7286#  7287#  7290#  7291   7293#  7294#  7296#  7297   7299#  7300#  7302#  7303
7308#  7309#  7313#  7314   7316#  7317#  7319#  7320#  7344#  7345   7347#  7348   7350#
7351   7353#  7354   7356#  7357   7359#  7360   7362#  7363   7365#  7366   7368#  7369
```

7374# 7375 7377# 7378 7380# 7381 7383# 7384 7386# 7387 7389# 7390 7392#
7393 7395# 7396 7398# 7399 7401# 7402 7404# 7405 7407# 7408 7410# 7411#
7413# 7414 7419# 7420 7422# 7423# 7424# 7426 7428# 7429 7431# 7432 7434#
7435 7437# 7438 7441# 7442# 7445# 7446# 7447# 7448# 7450# 7451 7456# 7457
7459# 7460# 7462# 7463 7465# 7466 7468# 7469 7471# 7472 7475# 7476 7478#
7479# 7481# 7485# 7486 7488# 7489# 7491# 7492# 7494 7518# 7519 7521# 7522
7524# 7525 7527# 7528# 7530# 7531# 7533# 7534 7536# 7537 7539# 7540 7542#
7543 7545# 7546 7569# 7570 7572# 7573# 7574# 7576# 7579 7581# 7582 7584#
7585 7587# 7588# 7590# 7591 7593# 7594 7596# 7597 7599# 7600 7602# 7603
7605# 7606 7608# 7609 7612# 7613 7614# 7615 7617# 7618# 7623# 7624 7626#
7627 7629# 7630 7642# 7643 7644# 7645# 7646 7648# 7650# 7651 7654# 7655 7657# 7658# 7660# 7661# 7663 7664# 7665# 7666 7668#
7669 7671# 7672 7673# 7674# 7677 7678# 7700# 7702# 7703 7705# 7706 7708# 7709 7711#
7690# 7691 7696# 7697 7698# 7699 7701# 7703 7705# 7706 7708# 7709 7711#
7712 7714# 7715 7718# 7719 7721# 7725# 7727# 7728# 7729 7731# 7732#
7734# 7736# 7761# 7762# 7764# 7765# 7767# 7768 7770# 7771 7773# 7774 7776#
7777 7779# 7780 7782# 7783 7784# 7786# 7787# 7788# 7791# 7792 7794# 7795
7797# 7798 7800# 7801 7803# 7804# 7806# 7807# 7808# 7810 7812# 7813 7815#
7816 7818# 7819 7821# 7822# 7824# 7826# 7827# 7828# 7830# 7831 7833# 7834#
7836# 7837# 7839# 7840 7841# 7843# 7845# 7847 7848# 7850# 7851 7853# 7855#
7856 7858# 7859 7861# 7862# 7864# 7865# 7866# 7868# 7871# 7872# 7878# 7899
7901# 7902 7904# 7905 7907# 7908# 7910# 7911# 7913# 7914 7916# 7917 7919#
7920 7922# 7923 7925# 7926# 7928# 7929# 7930# 7932# 7934# 7935 7937# 7938
7940# 7941 7943# 7944 7947# 7948# 7950# 7951# 7953# 7954 7956# 7959# 7961#
7962 7964# 7965# 7967# 7968# 7970# 7971 7973# 7974 7976# 7977 7979# 7980
7982# 7983 7985# 7986 7988# 7989# 7991# 7992# 7994 7996# 7997 7998 8000#
8001# 8003# 8004# 8006# 8007# 8009# 8010 8012# 8013# 8015# 8016 8018# 8019
8021# 8022 8024# 8025# 8027# 8028# 8030# 8031 8033# 8034 8036# 8037 8039#
8040 8042# 8043# 8045# 8046# 8048# 8049# 8051# 8054# 8057# 8058# 8060# 8061
8063# 8064# 8066# 8067# 8069# 8070 8072# 8073 8075# 8077 8079# 8080# 8082#
8083# 8085# 8086# 8088# 8089# 8091# 8092# 8094# 8097# 8098 8100# 8101#
8104# 8105 8107# 8108# 8110# 8111# 8113# 8114# 8140 8142# 8145# 8146 8148#
8149 8151# 8152 8154# 8155# 8157# 8158# 8160# 8161 8165 8167# 8168
8170# 8171 8176# 8177 8179# 8180# 8182# 8183# 8185# 8186# 8189# 8191#
8192 8194# 8195 8197# 8198# 8200# 8201# 8203# 8204# 8206# 8207# 8210# 8211#
8214# 8215 8217# 8218# 8220# 8221# 8223# 8224# 8226# 8227# 8233# 8235#
8236 8238# 8239# 8241# 8242# 8244# 8245# 8247# 8248# 8250# 8251# 8257# 8258
8260# 8261# 8263# 8264# 8266# 8267# 8269# 8270# 8272# 8273# 8300 8302#
8303 8305# 8306# 8308# 8309# 8311# 8312# 8314# 8315# 8317# 8318 8321# 8322
8324# 8325# 8327# 8328# 8330# 8331# 8333# 8334# 8336# 8337# 8343 8345#
8346 8348# 8349# 8351# 8352# 8354# 8355# 8357# 8358# 8360# 8361# 8365
8367# 8368# 8370# 8371# 8373# 8374# 8376# 8377# 8379# 8380# 8384 8389#
8390# 8392# 8393# 8395# 8396# 8398# 8399# 8401# 8402# 8404# 8405# 8410# 8411#
8414# 8415 8417# 8418# 8420# 8421# 8423# 8424# 8426# 8427# 8431# 8456#
8457 8459# 8460# 8462# 8463# 8465# 8466# 8468# 8469# 8471# 8474# 8475
8477# 8478 8480# 8481# 8483# 8484# 8486# 8487# 8489# 8492# 8496# 8498#
8499 8501# 8502# 8504# 8505# 8507# 8508# 8510# 8511# 8512 8515# 8517# 8518#
8521# 8522 8524# 8525# 8527# 8528# 8530# 8531 8533# 8534 8537 8542#
8543 8545# 8546# 8548# 8549# 8551# 8552# 8554# 8555# 8558# 8559 8561# 8562
8564# 8565 8567# 8568# 8570# 8571# 8573# 8574 8576# 8577# 8580 8583 8585#
8586 8588# 8589 8591# 8592 8595# 8598# 8599# 8601# 8602# 8605# 8606
8608# 8609# 8611# 8612# 8614# 8615 8617# 8618 8620# 8621 8627 8629#
8630# 8632# 8633# 8637# 8638 8640# 8641# 8643# 8644# 8669# 8670 8672# 8673
8675# 8676 8678# 8679 8681# 8682 8684# 8685 8687# 8688 8690# 8691 8693#
8694 8696# 8697 8702# 8703 8705# 8706# 8708# 8709# 8712# 8713 8715# 8716

9991 9993# 9994 9996# 9997 9999# 10000 10002# 10003 10005# 10006 10008# 10009
10011# 10012 10014# 10015 10017# 10018 10020# 10021 10023# 10024 10026# 10027 10029#
10030 10032# 10033 10035# 10036 10039# 10040 10043# 10044 10046# 10047 10049# 10050
10052# 10053 10055# 10056 10058# 10061# 10062# 10064# 10065# 10067# 10068# 10070# 10071 10073#
10074 10076# 10077 10079# 10080 10083# 10084 10085# 10087# 10089# 10090 10092# 10093#
10096# 10097 10099# 10100# 10102# 10103 10105# 10106# 10108# 10109 10111# 10112 10114#
10115 10117# 10118 10120# 10121 10123# 10124 10126# 10130 10122# 10133# 10135# 10136
10138# 10139 10141# 10142 10144# 10145 10147# 10148 10150# 10151 10154# 10155 10157#
10158# 10160# 10161 10163# 10164# 10167# 10168 10170# 10171# 10173# 10174 10176# 10177#
10179# 10180 10182# 10184# 10185 10187# 10188# 10190# 10191# 10194# 10195 10197# 10198#
10200# 10201# 10203# 10204# 10206# 10207# 10209# 10211# 10213# 10214# 10216# 10229 10241# 10242 10244#
10245 10247# 10248# 10250# 10251 10253# 10254# 10256# 10257 10259# 10260 10262# 10263
10265# 10266 10268# 10269 10271# 10272 10274# 10275 10277# 10278 10280# 10281 10283#
10284 10286# 10287 10289# 10290 10292# 10293 10295# 10296 10298# 10299 10301# 10302
10304# 10305 10307# 10308 10310# 10311 10313# 10314 10316# 10317 10319# 10320 10322#
10323 10325# 10326 10328# 10329 10331# 10332 10334# 10335 10337# 10338 10340# 10341
10343# 10344 10346# 10347 10349# 10350 10352# 10353 10355# 10356 10357 10360# 10361 10363#
10364 10366# 10367 10369# 10370 10373# 10374 10376# 10377 10379# 10381# 10382 10384# 10385
10387# 10388 10390# 10391 10393# 10394 10396# 10397 10399# 10400# 10401 10403# 10404# 10406#
10407 10409# 10410# 10412# 10414 10416# 10417# 10419# 10420 10422# 10423# 10425# 10426
10428# 10429 10431# 10432 10434# 10435 10437# 10438 10440# 10441 10443# 10446# 10447 10449#
10450# 10452# 10453 10455# 10456 10458# 10459 10461# 10462 10464# 10465 10467# 10468
10471# 10472 10474# 10475 10477# 10478 10480# 10481# 10482 10484# 10485 10487# 10488# 10490#
10491 10493# 10494# 10496# 10497 10499# 10501# 10502 10504# 10505# 10507# 10508# 10511#
10512 10514# 10515# 10517# 10518# 10520# 10521# 10522 10524# 10525 10527# 10528 10531#
10532# 10534# 10535 10537# 10538# 10540# 10541 10543# 10544# 10546# 10547 10549# 10550
10553# 10554 10556# 10557 10559# 10560 10562# 10563 10565# 10566 10568# 10569 10571 10573# 10574 10576#
10577 10579# 10580 10582# 10583 10585# 10586 10588# 10589 10591# 10592 10594# 10595
10597# 10598 10600# 10601 10603# 10604 10606# 10607 10609# 10610 10612# 10613 10615#
10616 10618# 10619 10621# 10622 10624# 10625 10627# 10628 10630# 10631 10633# 10634
10636# 10637 10639# 10640 10642# 10643 10645# 10646 10648# 10649 10651# 10652 10654#
10655 10657# 10658 10660# 10661 10663# 10664 10666# 10667 10669# 10670 10673# 10674
10677# 10678 10680# 10681 10683# 10684 10686# 10687 10689# 10693 10695# 10696# 10698#
10699 10701# 10702 10704# 10705 10707# 10708 10710# 10711 10713# 10714 10717# 10718
10720# 10721# 10723# 10724 10726# 10727# 10730# 10731 10733# 10734# 10736# 10737 10739#
10740# 10742# 10743 10745# 10746 10748# 10749 10751# 10752 10754# 10755 10757# 10758
10763# 10764 10766# 10767# 10769# 10770 10772# 10773 10775# 10776 10778# 10779 10781#
10782 10784# 10785 10787# 10788 10790# 10791# 10792# 10795 10797# 10798# 10801# 10802
10804# 10805# 10807# 10808 10810# 10811# 10813# 10814 10816# 10818# 10819 10821# 10822#
10824# 10825# 10828# 10829 10831# 10832# 10834# 10835# 10838# 10839 10841# 10842 10844#
10870 10872# 10873 10875# 10876 10878# 10879 10881# 10882 10884# 10885 10887# 10888
10890# 10891 10893# 10894 10896# 10897 10899# 10900 10902# 10903 10905# 10906 10908#
10909 10911# 10912 10914# 10915 10917# 10918 10920# 10921 10923# 10924 10926# 10927
10929# 10930 10932# 10933 10935# 10936 10938# 10939 10941# 10942 10944# 10945 10947#
10948 10950# 10951 10953# 10954 10956# 10957 10959# 10960 10962# 10963 10965# 10966
10968# 10969 10971# 10972 10974# 10975 10977# 10978 10980# 10981 10983# 10984 10986#
10987 10990# 10991 10994# 10995 10997# 10998 11000# 11001 11003# 11004 11009# 11010
11012# 11013# 11015# 11016 11018# 11019 11021# 11022 11024# 11025 11027# 11028 11030#
11031 11034# 11035 11037# 11038# 11040# 11041 11043# 11044# 11047# 11048 11050# 11051#
11053# 11054 11056# 11057# 11059# 11060 11062# 11063 11065# 11066 11068# 11069 11071#
11072 11074# 11075 11080# 11081 11083# 11084# 11086# 11087 11089# 11090 11092# 11093
11095# 11096 11098# 11099 11101# 11102 11105# 11106 11108# 11109# 11111# 11112 11114#
11115# 11118# 11119 11121# 11122# 11124# 11125 11127# 11128# 11130# 11131 11133# 11135#
11136 11138# 11139# 11141# 11142# 11145# 11146 11148# 11149# 11151# 11152# 11180# 11181
11183# 11184 11186# 11187 11189# 11190 11192# 11193 11195# 11196 11198# 11199 11201#
11202 11204# 11205 11207# 11208 11210# 11211 11213# 11214 11216# 11217 11219# 11220
11222# 11223 11225# 11226 11228# 11229 11231# 11232 11234# 11235 11237# 11238 11240#



F12

12491 12493# 12494 12496# 12497 12499# 12500 12502# 12503 12505# 12506 12508# 12509
12511# 12512 12514# 12515 12517# 12518 12520# 12521 12523# 12524 12526# 12527 12529#
12530 12532# 12533 12535# 12536 12538# 12539 12541# 12542 12544# 12545 12547# 12548
12550# 12551 12553# 12554 12556# 12557 12559# 12560 12562# 12563 12565# 12566 12568# 12569
12569 12571# 12572 12574# 12575 12576 12578# 12580 12581 12583# 12584 12586# 12589
12594# 12595 12597# 12598# 12600# 12601 12603# 12604 12606# 12607 12609# 12610 12612#
12613 12615# 12616 12618# 12619 12621# 12622 12624# 12625 12627# 12628 12630# 12633
12635# 12636# 12638# 12639# 12641# 12642# 12643# 12645# 12646# 12648# 12649# 12651 12653#
12654 12656# 12657 12659# 12660 12662# 12663# 12665# 12666# 12668# 12669# 12671# 12672 12674# 12675
12677# 12679 12680# 12681 12683# 12684# 12686# 12687 12689# 12690# 12691 12693# 12694# 12696#
12697 12699# 12700# 12703# 12704 12706# 12707 12709# 12710 12712# 12713# 12715# 12716
12718# 12720# 12721 12723# 12724# 12726# 12727 12729# 12730 12731 12733# 12734# 12736# 12737#
12765# 12766 12768# 12769 12771# 12772# 12773 12775# 12776 12777 12779# 12780# 12781 12783#
12784 12786# 12787 12789# 12790 12792# 12793 12795# 12796 12797 12799# 12801# 12802
12804# 12805 12807# 12808 12810# 12811 12813# 12814 12816# 12817 12819# 12820 12822#
12823 12825# 12826 12828# 12829 12831# 12832 12834# 12835 12837# 12838 12840# 12841
12843# 12844 12846# 12847 12849# 12850 12852# 12853 12855# 12856 12858# 12859 12861#
12862 12864# 12865 12867# 12868 12870# 12871 12873# 12874 12876# 12877 12879# 12880
12882# 12883 12885# 12886 12888# 12889 12891# 12892 12894# 12895 12897# 12898# 12900 12902#
12903 12905# 12906 12911# 12912 12914# 12915 12917# 12918 12920# 12921 12923# 12924
12926# 12927 12928# 12930 12932# 12933 12935# 12936 12938# 12939 12941# 12942# 12943 12945#
12946# 12948# 12949# 12951 12952# 12954# 12955 12957# 12958 12960# 12961 12963# 12964# 12965
12967# 12968 12970# 12971 12974# 12976# 12977 12979# 12980 12982# 12983 12985# 12986# 12988#
12989 12991# 12992 12994# 12995 12997# 12998 13000# 13001 13003# 13004 13007# 13008
13010# 13011# 13013# 13014 13016# 13017# 13020# 13021 13023# 13024# 13026# 13027 13029#
13030# 13032# 13033 13035# 13036# 13038 13039# 13041# 13043# 13044# 13047# 13048 13050#
13051# 13053# 13054# 13056# 13057 13059# 13060# 13062# 13063# 13065# 13066# 13068# 13069#
13097# 13098 13100# 13101 13103# 13104 13106# 13107 13109# 13110 13112# 13113 13115#
13116 13118# 13119 13121# 13122 13124# 13125 13127# 13128 13130# 13131 13133# 13134
13136# 13137 13139# 13140 13142# 13143 13145# 13146 13148# 13149 13151# 13152 13154#
13155 13157# 13158 13160# 13161 13163# 13164 13166# 13167 13169# 13170 13172# 13173
13175# 13176 13178# 13179 13181# 13182 13184# 13185 13187# 13188 13190# 13191 13193#
13194 13196# 13197 13199# 13200 13202# 13203 13205# 13206 13209# 13210 13213# 13214
13216# 13217 13219# 13220 13222# 13223 13225# 13226 13228# 13229 13234# 13235 13237#
13238 13240# 13241 13243# 13244 13246# 13247 13249# 13250 13253# 13254 13256# 13257#
13259# 13260 13262# 13263# 13265# 13266# 13268# 13269# 13270# 13272# 13273 13275# 13276# 13278#
13279 13281# 13282 13284# 13285 13287# 13288 13290# 13291 13293# 13294 13299# 13300
13302# 13303# 13305# 13306 13308# 13309# 13311# 13312 13314# 13315 13317# 13318 13320#
13321 13324# 13325 13327# 13328# 13330# 13331 13333# 13334# 13337# 13338 13340# 13341#
13343# 13344 13346# 13347# 13349# 13350# 13352# 13354# 13355 13357# 13358# 13360# 13361#
13364# 13365 13367# 13368# 13370# 13371# 13373# 13400 13402# 13403 13405# 13406 13408#
13409 13411# 13412 13414# 13415 13417# 13418 13420# 13421 13423# 13424 13426# 13427
13429# 13430 13432# 13433 13435# 13436# 13438# 13439 13441# 13442 13444# 13445 13447#
13448 13450# 13451 13453# 13454# 13456# 13457 13459# 13460 13462# 13463 13465# 13466
13468# 13469 13471# 13472 13474# 13475# 13477# 13478 13480# 13481 13483# 13484# 13486#
13487 13489# 13490 13492# 13493 13495# 13496 13498# 13499 13501# 13502 13504# 13505
13507# 13508 13510# 13511 13513# 13514 13516# 13517 13519# 13520 13522# 13523 13526#
13527 13530# 13531 13533# 13534 13536# 13537 13540# 13540 13543# 13544 13546# 13547 13549#
13551# 13552 13554# 13555 13557# 13560# 13562# 13563# 13564 13566# 13567 13570#
13571 13573# 13574# 13576# 13577 13579# 13580# 13582# 13583# 13584# 13587# 13589# 13590
13592# 13593# 13595# 13596 13598# 13599 13601# 13602 13604# 13605 13607# 13608 13610#
13611 13616# 13617 13619# 13620# 13622# 13623 13625# 13626 13628# 13629 13631# 13632
13634# 13635 13637# 13638 13641# 13642 13644# 13645# 13647# 13648 13650# 13651# 13654#
13655 13657# 13658# 13660# 13661 13663# 13664# 13666# 13667 13669# 13671# 13672 13674#
13675# 13677# 13678# 13681# 13682 13684# 13685# 13687# 13688# 13714# 13715 13717# 13718
13720# 13721 13723# 13724 13726# 13727 13729# 13730 13732# 13733# 13735# 13736 13738#

```
13739   13742#  13743   13745#  13746#  13748#  13749#  13751#  13752   13754#  13755   13757#  13758
13760#  13761#  13763#  13764#  13767#  13768   13770#  13771   13774#  13775   13777#  13778#  13780#
13781#  13783#  13784   13786#  13787#  13789#  13790   13792#  13793   13795#  13796   13798#  13799#
13801#  13802#  13805#  13806   13808#  13809   13812#  13813   13815#  13816#  13818#  13819#  13822#
13823   13825#  13826#  13828#  13829#  13832#  13834#  13836#  13838#  13841   13842#  13846#  13847
13870#  13871   13873#  13874   13876#  13877   13878#  13880   13882#  13883   13885#  13886   13888#
13889   13892#  13893   13895#  13896   13900#  13901   13903#  13904   13905#  13907   13909#  13910
13912#  13913   13915#  13916   13918#  13919   13921#  13922   13923#  13926#  13930#  13931   13933#
13934   13936#  13937   13939#  13940   13942#  13943   13945#  13946   13948#  13949   13951#  13952
13954#  13955   13957#  13958   13962#  13963   13965#  13966#  13967#  13970#  13971#  13972   13974#
13975   13980#  13981   13983#  13984#  13986#  13987#  13990#  13991   13993#  13994#  13996#  13997
13999#  14000#  14003#  14004   14006#  14007#  14009#  14010   14012#  14013   14015#  14016#  14018#
14019#  14021#  14022   14024#  14025#  14027#  14028   14031#  14032   14034#  14035   14037#  14038
14040#  14041#  14046#  14047   14049#  14050   14053#  14054   14056#  14057#  14059#  14060   14063#
14064   14066#  14067#  14069#  14070   14073#  14074   14076#  14077#  14079#  14080   14083#  14084
14086#  14087#  14089#  14090   14093#  14094   14096#  14097#  14099#  14100   14103#  14104   14106#
14107#  14109#  14110   14113#  14114   14116#  14117#  14121#  14122   14125#  14126   14128#  14129#
14131#  14132   14135#  14136   14138#  14139#  14141#  14142   14145#  14146   14148#  14149#  14151#
14152   14155#  14156   14158#  14159#  14161#  14162   14165#  14166   14168#  14169#  14171#  14172
14175#  14176   14178#  14179#  14181#  14182   14186#  14187   14188#  14189#  14191#  14192   14195#
14196   14198#  14199#  14204#  14205   14207#  14208   14211#  14212   14214#  14215#  14217#  14218
14220#  14221   14223#  14224   14226#  14227   14230#  14231   14233#  14234#  14236#  14237   14239#
14240   14243#  14244   14246#  14247#  14249#  14250   14252#  14253   14256#  14257   14259#  14260#
14262#  14263   14265#  14266   14268#  14270   14272#  14273#  14275#  14276   14278#  14279   14282#
14283   14285#  14286#  14288#  14289   14291#  14292   14295#  14296   14298#  14299#  14301#  14302
14304#  14305   14308#  14309   14311#  14312#  14314#  14315   14317#  14318   14321#  14322   14324#
14325#  14327#  14328   14330#  14331   14334#  14335   14337#  14338#  14343#  14344   14346#  14347
14349#  14350   14352#  14353   14355#  14356   14361#  14362   14364#  14365#  14367#  14368   14370#
14371   14373#  14374   14376#  14377   14380#  14381   14383#  14384#  14386#  14387#  14389#  14390
14392#  14393#  14395#  14396#  14398#  14399   14401#  14402#  14404#  14405#  14408#  14409   14411#
14412#  14414#  14415#  14417#  14418   14420#  14421   14423#  14427   14429#  14430#  14432#  14433
14435#  14436   14438#  14439   14441#  14442   14446#  14447   14448#  14449   14451#  14452#  14455#
14456#  14461#  14462   14464#  14465   14467#  14468   14470#  14471   14473#  14474   14476#  14477   14482
14499#  14500   14502#  14503   14505#  14506   14508#  14509   14511#  14512   14514#  14515   14517#
14518   14520#  14521   14523#  14524   14526#  14527   14530#  14531   14532#  14533   14535#  14536
14538#  14539   14541#  14542   14544#  14545   14547#  14548#  14550#  14551   14553#  14554   14556#
14557   14559#  14560#  14562#  14563   14565#  14566   14568#  14569   14571#  14572#  14574#  14575
14577#  14578#  14580#  14581   14583#  14584   14586#  14587#  14589#  14590   14592#  14593#  14595#
14596   14598#  14599   14601#  14602#  14604#  14605   14607#  14608   14613#  14614   14616#  14617#
14619#  14620#  14623#  14624   14626#  14627   14632#  14633   14635#  14636#  14638#  14639   14641#
14642   14644#  14645   14647#  14648   14650#  14651   14654#  14655   14657#  14658#  14660#  14661#
14664#  14665   14667#  14668#  14670#  14671#  14694#  14700   14702#  14703   14705#  14706   14708#
14709   14711#  14712   14714#  14715   14717#  14718   14720#  14721   14723#  14724   14726#  14727
14729#  14730   14732#  14733   14735#  14736#  14738#  14739   14741#  14742#  14747#  14748   14750#
14751   14753#  14754   14758#  14759   14761#  14762#  14764#  14765#  14767#  14768   14770#  14771
14773#  14774   14776#  14777#  14779#  14780#  14784#  14785   14787#  14788#  14809#  14810   14812#
14813   14815#  14816   14818#  14819   14821#  14822   14824#  14825   14827#  14828   14830#  14831
14833#  14834   14836#  14837   14839#  14840   14842#  14843   14845#  14846   14848#  14849#  14851#
14852   14854#  14855#  14857#  14858#  14860#  14861   14863#  14864   14866#  14867   14869#  14870
14872#  14873   14875#  14876#  14885#  14886   14888#  14902   14891#  14892   14894#  14895   14897#
14898   14900#  14901   14903#  14904   14906#  14907   14909#  14910   14912#  14913   14915#  14916
14918#  14919#  14921#  14922   14924#  14925#  14927#  14928#  14930#  14931   14933#  14934   14936#
14937   14939#  14940   14942#  14943   14945#  14946#  14955#  14956   14958#  14959   14961#  14962
14964#  14965   14967#  14968   14970#  14971   14973#  14974   14976#  14977   14979#  14980   14982#
14983#  14985#  14986   14988#  14989   14991#  14992   14994#  14995   14997#  14998#  15000#  15001#
15010#  15011   15013#  15014   15016#  15017   15019#  15020   15022#  15023   15025#  15026   15028#
```

```
                    15029   15031#  15032#  15034#  15035   15037#  15038#  15041#  15042   15044#  15045#  15047#  15048
                    15050#  15051#
.2A    = 000120       1#   12588   12656
.2AWC  = 000140       1#    5159    5229   11003   11071
```

```
ADDMC       1#  11636  11704
ADDI        1#
AND         1#   3091   3180   3308   3406   7860   9734   9802  14534  14835  14905  14969  14990
BBO         1#   7409   7616   7643   7826   7987   8870   8921  14570  14853  15036
BBI         1#  14576  14591  14923
BBW         1#  13731  13759  13785  14558  14740
BBT         1#   2361   2441   2558   2638   2755   2835   2952   3032   3159   3248   3385   3483   3600   3680
             3797   3877   3994   4074   4191   4271   4388   4468   4585   4665   4782   4862   4979   5059   5167
             5237   5359   5444   5566   5651   5773   5858   5980   6065   6187   6272   6394   6479   6601   6686
             6808   6893   7015   7100   7222   7307   8231   8388  13797  14546  14734  14760  14847  14856  14917
             14926  14981  14996  15030
BC          1#   7832   8249   8409   8544   8628   8704   8794  13982  14014  14391  14775
BZ          1#   2308   2388   2505   2585   2702   2782   2899   2979   3106   3195   3332   3430   3547   3627
             3744   3824   3941   4021   4138   4218   4335   4415   4532   4612   4729   4809   4926   5006   5123
             5193   5303   5388   5510   5595   5717   5802   5924   6009   6131   6216   6338   6423   6545   6630
             6752   6837   6959   7044   7166   7251   7421   7458   7658   7698   7964   7999   8056   8087   8178
             8335   8485   8575   8735   9109   9180   9236   9426   9497   9552   9743   9814   9869  10060  10131
            10186  10377  10448  10503  10694  10765  10820  11011  11082  11137  11328  11399  11454  11645  11716
            11771  11962  12033  12088  12279  12350  12405  12596  12667  12722  12913  12984  13039  13230  13301
            13356  13547  13618  13673  14005  14363  14400  14428  14615  14634
CALL        1#   2329   2342   2409   2422   2445   2526   2539   2606   2619   2642   2723   2736   2803   2816
             2839   2920   2933   3000   3013   3036   3127   3140   3216   3229   3252   3353   3366   3451   3464
             3487   3568   3581   3648   3661   3684   3765   3778   3845   3858   3881   3962   3975   4042   4055
             4078   4159   4172   4239   4252   4275   4356   4369   4436   4449   4472   4553   4566   4633   4646
             4669   4750   4763   4830   4843   4866   4947   4960   5027   5040   5063   5135   5145   5205   5215
             5241   5327   5340   5412   5425   5448   5534   5547   5619   5632   5655   5741   5754   5826   5839
             5862   5948   5961   6033   6046   6069   6155   6168   6240   6253   6276   6362   6375   6447   6460
             6483   6569   6582   6654   6667   6690   6776   6789   6861   6874   6897   6983   6996   7068   7081
             7104   7190   7203   7275   7288   7311   7439   7473   7483   7676   7716   7726   7844   7863   8074
             8102   8199   8212   8262   8356   8369   8422   8509   8519   8593   8603   8635   8753   8766   8801
             8880   8890   8924   9130   9143   9201   9214   9241   9247   9460   9518   9531   9558   9764   9777
             9835   9948   9975  10081  10094  10152  10165  10192  10398  10411  10469  10482  10589  10715  10728
            10786  10799  10826  11032  11045  11103  11116  11143  11349  11362  11420  11433  11460  11666  11679
            11737  11750  11777  11983  11996  12054  12067  12094  12300  12313  12371  12384  12411  12617  12630
            12688  12701  12728  12934  12947  13005  13018  13045  13251  13264  13322  13335  13362  13568  13581
            13639  13652  13679  13740  13772  13810  13820  14051  14061  14071  14081  14091  14101  14111  14123
            14133  14143  14153  14163  14173  14183  14193  14209  14228  14241  14254  14267  14280  14293  14306
            14319  14332  14378  14406  14446  14652  14662  15039
COMMEM    176#
DISP        1#
ENDCOM    176#
ERROR      70#   2277   2474   2671   2868   3066   3283   3516   3713   3910   4107   4304   4501   4698   4895
             5093   5270   5477   5684   5891   6088   6305   6512   6719   6926   7133   7339   7513   7756   7893
             8133   8290   8451   8664   8830   8956   9273   9590   9907  10224  10541  10858  11175  11492  11809
            12126  12443  12760  13077  13394  13709  13852  14476  14694  15085  15119  15135  15142
ESCAPE    176#
GETPRI    176#
GETSWR    176#
HLT       189#
KMEND       1#   953
KMFRNT      1#    30
LORN        1#  9417   9485
MOVE        1#  2282   2285   2289   2293   2296   2299   2311   2314   2317   2320   2323   2326   2330   2336
             2343   2349   2355   2369   2373   2376   2379   2391   2394   2397   2400   2403   2406   2410   2416
             2423   2429   2435   2446   2479   2482   2486   2490   2493   2496   2508   2511   2514   2517   2520
             2523   2527   2533   2540   2546   2552   2566   2570   2573   2576   2588   2591   2594   2597   2600
```

2603	2607	2613	2620	2626	2632	2643	2676	2679	2683	2687	2690	2693	2705	2708
2711	2714	2717	2720	2724	2730	2737	2743	2749	2763	2767	2770	2773	2785	2788
2791	2794	2797	2800	2804	2810	2817	2823	2830	2840	2873	2876	2880	2884	2887
2890	2902	2905	2908	2911	2914	2917	2921	2927	2934	2940	2946	2960	2964	2967
2970	2982	2985	2988	2991	2994	2997	3001	3007	3014	3020	3026	3037	3071	3074
3078	3082	3085	3088	3094	3097	3109	3112	3115	3118	3121	3124	3128	3134	3141
3147	3153	3167	3171	3174	3177	3183	3186	3198	3201	3204	3207	3210	3213	3217
3223	3230	3236	3242	3253	3288	3291	3295	3299	3302	3305	3311	3314	3317	3320
3335	3338	3341	3344	3347	3350	3354	3360	3362	3373	3379	3383	3397	3400	3403
3409	3412	3415	3418	3433	3436	3439	3442	3445	3448	3452	3458	3465	3471	3477
3488	3521	3524	3528	3532	3536	3538	3550	3553	3556	3559	3562	3565	3569	3575
3582	3588	3594	3608	3612	3615	3618	3630	3633	3638	3643	3646	3649	3649	3655
3662	3668	3674	3685	3718	3721	3725	3729	3732	3735	3747	3750	3753	3756	3759
3762	3766	3772	3779	3785	3791	3805	3809	3812	3815	3827	3830	3833	3836	3839
3842	3846	3852	3859	3865	3871	3882	3915	3918	3922	3926	3928	3932	3944	3947
3950	3963	3966	3969	3973	3979	3996	3999	4002	4005	4006	4009	4012	4024	4027
4030	4033	4036	4039	4043	4049	4056	4062	4068	4079	4112	4115	4119	4123	4126
4129	4141	4144	4147	4150	4153	4156	4160	4166	4173	4179	4185	4199	4203	4206
4209	4221	4224	4227	4230	4233	4236	4240	4246	4253	4259	4265	4276	4309	4312
4316	4320	4323	4326	4338	4341	4344	4347	4350	4353	4357	4363	4370	4376	4382
4396	4400	4403	4406	4418	4421	4424	4427	4430	4433	4437	4443	4450	4456	4462
4473	4506	4509	4513	4517	4520	4523	4535	4538	4541	4544	4547	4550	4554	4560
4567	4573	4579	4593	4597	4600	4603	4615	4618	4621	4624	4627	4630	4634	4640
4647	4653	4659	4670	4703	4706	4710	4714	4717	4720	4732	4735	4738	4741	4744
4747	4751	4757	4764	4770	4776	4790	4794	4797	4900	4812	4815	4818	4821	4824
4827	4831	4837	4844	4850	4856	4867	4900	4903	4907	4911	4914	4917	4929	4932
4935	4938	4941	4944	4948	4954	4961	4967	4973	4987	4991	4994	4997	5009	5012
5015	5018	5021	5024	5028	5034	5041	5047	5053	5064	5098	5101	5105	5108	5111
5114	5126	5129	5132	5136	5146	5155	5164	5175	5178	5181	5184	5196	5199	5202
5206	5216	5225	5234	5242	5275	5278	5282	5286	5288	5291	5294	5306	5309	5312
5315	5318	5321	5324	5328	5334	5341	5347	5363	5367	5370	5373	5376	5379	5391
5394	5397	5400	5403	5406	5409	5413	5419	5428	5432	5438	5449	5482	5485	5489
5492	5495	5498	5501	5513	5516	5519	5522	5588	5598	5531	5535	5541	5548	5554
5560	5574	5577	5580	5583	5588	5598	5601	5604	5607	5610	5613	5616	5620	5626
5633	5639	5645	5656	5669	5692	5698	5699	5702	5705	5708	5720	5723	5726	5729
5732	5735	5738	5742	5748	5755	5761	5767	5781	5784	5787	5790	5793	5805	5808
5811	5814	5817	5820	5823	5827	5833	5840	5846	5852	5863	5996	5999	5903	5906
5909	5912	5915	5927	5930	5933	5936	5939	5942	5945	5949	5955	5962	5968	5974
5988	5991	5994	5997	6000	6012	6015	6018	6021	6024	6027	6030	6034	6040	6047
6053	6059	6070	6103	6106	6110	6113	6116	6119	6122	6134	6137	6140	6143	6146
6149	6152	6156	6162	6169	6175	6181	6185	6190	6201	6204	6207	6219	6222	6225
6228	6231	6234	6237	6241	6247	6254	6260	6266	6277	6310	6313	6317	6320	6323
6326	6329	6341	6344	6347	6350	6353	6356	6359	6363	6369	6376	6382	6388	6402
6405	6408	6411	6414	6426	6429	6432	6435	6438	6441	6444	6448	6454	6461	6467
6473	6484	6517	6520	6524	6527	6530	6533	6536	6548	6551	6554	6557	6560	6563
6566	6570	6576	6583	6589	6595	6609	6612	6615	6618	6621	6633	6636	6639	6642
6645	6648	6651	6655	6661	6668	6674	6680	6691	6724	6727	6731	6734	6737	6740
6743	6755	6758	6761	6764	6767	6770	6773	6777	6783	6790	6796	6802	6816	6819
6822	6825	6828	6840	6843	6846	6849	6852	6855	6858	6862	6868	6875	6881	6887
6898	6931	6934	6938	6941	6944	6947	6950	6962	6965	6968	6971	6974	6977	6980
6984	6990	6997	7003	7009	7023	7026	7029	7032	7035	7047	7050	7053	7056	7059
7062	7065	7069	7075	7082	7088	7094	7105	7138	7141	7145	7148	7151	7154	7157
7169	7172	7175	7178	7181	7184	7187	7191	7197	7204	7210	7216	7230	7233	7236
7239	7242	7254	7257	7260	7263	7266	7269	7272	7276	7282	7289	7295	7301	7312
7343	7346	7349	7352	7355	7358	7361	7364	7367	7373	7376	7379	7382	7385	7388
7391	7394	7397	7400	7403	7406	7412	7424	7427	7430	7433	7436	7440	7449	7461

```
7464  7467  7470  7474  7484  7517  7520  7523  7526  7529  7532  7535  7538  7541  7544
7547  7550  7556  7559  7565  7568  7571  7574  7577  7580  7583  7586  7589  7592  7595
7598  7601  7604  7607  7613  7622  7625  7628  7631  7634  7637  7640  7646  7649  7661
7663  7666  7669  7672  7675  7678  7681  7684  7687  7690  7693  7696  7699  7702  7705
7708  7711  7717  7720  7723  7730  7741  7745  7754  7757  7764  7797  7900  7903  7906
7909  7912  7915  7918  7921  7924  7927  7930  7933  7936  7939  7942  7945  7960  7963
7966  7969  7972  7975  7978  7981  7984  7988  7992  8014  8017  8020  8023  8026  8029
8032  8035  8038  8041  8044  8047  8059  8062  8065  8068  8071  8075  8103  8138  8141
8144  8147  8150  8153  8156  8159  8163  8166  8169  8181  8184  8187  8190  8193  8196
8200  8206  8213  8219  8225  8240  8243  8246  8263  8265  8268  8301  8304  8307  8310
8313  8316  8320  8323  8326  8338  8341  8344  8347  8350  8353  8357  8363  8370  8376
8382  8400  8413  8416  8423  8455  8458  8461  8464  8467  8470  8473  8476  8488  8491
8494  8497  8500  8503  8506  8510  8520  8535  8551  8554  8557  8560  8563  8566  8578
8581  8584  8587  8590  8594  8604  8619  8636  8640  8671  8674  8677  8680  8683  8686
8689  8711  8714  8717  8720  8723  8726  8730  8741  8744  8747  8750  8754  8760  8767
8773  8785  8802  8834  8837  8840  8843  8846  8849  8852  8855  8858  8867  8874  8877
8881  8891  8903  8906  8910  8923  8960  8963  8966  8969  8972  8975  8978  8984  8987
8990  8993  8996  8999  9002  9005  9008  9011  9014  9017  9020  9023  9026  9029  9032
9035  9038  9041  9044  9047  9050  9053  9056  9059  9062  9065  9068  9071  9074  9077
9080  9083  9087  9091  9097  9100  9112  9115  9118  9121  9124  9127  9131  9137  9144
9150  9156  9159  9162  9168  9171  9183  9186  9189  9192  9196  9198  9202  9208  9215
9221  9227  9242  9277  9280  9283  9286  9289  9292  9295  9301  9304  9307  9310  9313
9316  9319  9322  9325  9328  9331  9334  9337  9340  9343  9346  9349  9352  9355  9358
9361  9364  9367  9370  9373  9376  9379  9382  9385  9388  9391  9394  9397  9400  9404
9408  9414  9423  9432  9435  9438  9441  9444  9448  9451  9461  9467  9473  9476  9479
9488  9500  9503  9506  9509  9512  9515  9519  9525  9532  9538  9544  9559  9594  9597
9600  9603  9606  9609  9612  9618  9621  9624  9627  9630  9633  9636  9639  9642  9645
9648  9651  9654  9657  9660  9663  9666  9669  9672  9675  9678  9681  9684  9687  9690
9693  9696  9699  9702  9705  9708  9711  9714  9717  9721  9725  9731  9746  9749  9752
9755  9758  9761  9765  9771  9778  9794  9790  9793  9796  9805  9817  9820  9823  9826  9829
9829  9832  9836  9842  9845  9855  9861  9876  9911  9914  9917  9920  9923  9926  9929
9835  9838  9841  9844  9847  9850  9853  9856  9859  9862  9865  9868  9871  9874  9877
9880  9883  9886  9889  9892  9895  9898  9901  9904  9907  9910  9913  9916  9919  9922
10001  10004  10007  10010  10013  10016  10019  10022
10025  10028  10031  10034  10038  10042  10046  10049  10053  10056  10059  10062  10066  10069  10072  10075  10078  10082  10086
10095  10101  10107  10110  10113  10122  10134  10137  10140  10143  10146  10149  10153  10159  10166
10172  10178  10193  10228  10231  10234  10237  10240  10243  10246  10249  10252  10255  10258  10261  10264
10267  10270  10273  10276  10279  10282  10285  10288  10291  10294  10297  10300  10303  10306  10309
10312  10315  10318  10321  10324  10327  10330  10333  10336  10339  10342  10345  10348  10351  10355
10359  10365  10380  10383  10386  10389  10392  10395  10398  10401  10405  10412  10418  10424  10427  10430
10439  10451  10454  10457  10460  10463  10466  10469  10472  10476  10479  10483  10485  10535  10545  10548
10551  10554  10557  10560  10563  10566  10570  10573  10576  10578  10581  10584  10587  10590  10593  10596
10599  10602  10605  10608  10611  10614  10617  10620  10623  10626  10629  10632  10635  10638  10641
10644  10647  10650  10653  10656  10659  10662  10665  10668  10672  10676  10679  10697  10700  10703
10706  10709  10712  10716  10722  10729  10732  10735  10738  10741  10744  10747  10750  10768  10771  10774  10777
10780  10783  10787  10793  10800  10806  10812  10827  10862  10865  10868  10871  10874  10877  10880
10886  10889  10892  10895  10898  10901  10904  10907  10910  10913  10916  10919  10922  10925  10928
10931  10934  10937  10940  10943  10946  10949  10952  10955  10958  10961  10964  10967  10970  10973
10976  10979  10982  10985  10989  10993  10999  11014  11017  11020  11023  11026  11029  11033  11039
11046  11052  11058  11061  11064  11073  11085  11088  11091  11094  11097  11100  11104  11110  11117
11123  11129  11144  11179  11182  11185  11188  11191  11194  11197  11203  11206  11209  11212  11215
11218  11221  11224  11227  11230  11233  11236  11239  11242  11245  11248  11251  11254  11257  11260
11263  11266  11269  11272  11275  11278  11281  11284  11287  11290  11293  11296  11299  11302  11306
11310  11316  11331  11334  11337  11340  11343  11346  11350  11356  11363  11369  11375  11378  11381
11390  11402  11405  11408  11411  11414  11417  11421  11427  11434  11440  11446  11461  11496  11499
11502  11505  11508  11511  11514  11520  11523  11526  11529  11532  11535  11538  11541  11544  11547
```

```
11550  11553  11556  11559  11562  11565  11568  11571  11574  11577  11580  11583  11586  11589  11592
11595  11598  11601  11604  11607  11610  11613  11616  11619  11623  11627  11633  11640  11651  11654
11657  11660  11663  11667  11673  11680  11686  11692  11645  11698  11707  11719  11722  11725  11728
11731  11734  11738  11744  11751  11757  11763  11770  11813  11816  11819  11822  11825  11828  11831
11837  11840  11843  11846  11849  11852  11855  11858  11861  11864  11867  11870  11873  11876  11879
11882  11885  11888  11891  11894  11897  11900  11903  11906  11909  11912  11915  11918  11921  11924
11927  11930  11933  11936  11940  11944  11950  11965  11968  11971  11974  11977  11980  11984  11990
11997  12003  12009  12012  12015  12024  12036  12039  12042  12045  12048  12051  12055  12061  12068
12074  12080  12095  12130  12133  12136  12139  12142  12145  12148  12154  12157  12160  12163  12166
12169  12172  12175  12178  12181  12184  12187  12190  12193  12196  12199  12202  12205  12208  12211
12214  12217  12220  12223  12226  12229  12232  12235  12238  12241  12244  12247  12250  12253  12257
12261  12267  12282  12286  12289  12291  12294  12297  12301  12307  12314  12320  12326  12329  12332
12341  12363  12366  12369  12372  12375  12378  12382  12391  12397  12401  12405  12412  12447  12450
12453  12456  12459  12462  12465  12471  12474  12477  12480  12483  12486  12489  12492  12495  12498
12501  12504  12507  12510  12513  12516  12519  12522  12525  12528  12531  12534  12537  12540  12543
12546  12549  12552  12555  12558  12561  12564  12567  12570  12574  12578  12584  12599  12602  12605
12608  12611  12614  12618  12624  12631  12637  12643  12646  12649  12668  12670  12673  12676  12678
12682  12685  12689  12695  12702  12708  12714  12720  12764  12767  12770  12773  12776  12779  12782
12788  12791  12794  12797  12800  12803  12806  12809  12812  12815  12818  12821  12824  12827  12830
12833  12836  12839  12842  12845  12848  12851  12854  12857  12860  12863  12866  12869  12872  12875
12878  12881  12884  12887  12891  12895  12901  12916  12919  12962  12965  12968  12971  12974  12977
12948  12954  12960  12963  12966  12975  12987  12990  12993  12996  12999  13002  13006  13012  13019
13025  13031  13046  13081  13084  13087  13090  13093  13096  13099  13105  13108  13111  13114  13117
13120  13123  13126  13129  13132  13135  13138  13141  13144  13147  13150  13153  13156  13159  13162
13165  13168  13171  13174  13177  13180  13183  13186  13189  13192  13195  13198  13201  13204  13208
13212  13218  13233  13236  13239  13242  13245  13248  13252  13265  13268  13271  13277  13280  13283
13292  13304  13307  13310  13313  13316  13319  13322  13329  13336  13342  13348  13363  13398  13401
13404  13407  13410  13413  13416  13419  13422  13425  13431  13434  13437  13440  13443  13446  13449
13452  13455  13458  13461  13464  13467  13470  13473  13476  13479  13482  13485  13488  13491  13494
13497  13500  13503  13506  13509  13512  13515  13518  13521  13525  13529  13535  13550  13553  13556
13559  13562  13565  13568  13571  13574  13577  13580  13583  13586  13589  13592  13595  13597  13630
13633  13636  13640  13646  13653  13662  13665  13668  13713  13716  13719  13722  13725  13728  13734
13737  13741  13750  13753  13756  13763  13765  13773  13782  13794  13804  13807  13811  13821  13856
13859  13862  13865  13869  13875  13878  13881  13884  13887  13891  13894  13897  13902  13905  13953
13908  13911  13914  13917  13920  13926  13929  13932  13935  13938  13941  13944  13947  13950  13953
13956  13961  13970  13973  13989  13995  14000  14020  14026  14030  14033  14045  14048  14052  14058
14062  14068  14072  14078  14082  14088  14092  14098  14102  14108  14112  14120  14124  14130  14134
14140  14144  14150  14154  14160  14164  14170  14174  14180  14184  14190  14194  14203  14206  14210
14216  14219  14222  14225  14229  14235  14238  14242  14248  14251  14255  14261  14264  14268  14274
14277  14281  14287  14290  14294  14300  14303  14307  14313  14316  14320  14326  14329  14333  14342
14345  14348  14351  14354  14366  14369  14372  14375  14379  14407  14416  14419  14431  14434  14437
14440  14443  14447  14460  14483  14486  14489  14492  14495  14498  14501  14504  14507  14510  14513
14516  14519  14522  14525  14528  14531  14537  14543  14549  14552  14555  14561  14564  14567  14573
14579  14582  14588  14594  14597  14603  14606  14622  14625  14637  14640  14643  14646  14649  14653
14663  14698  14701  14704  14707  14710  14713  14716  14722  14725  14731  14737  14746  14749  14757
14783  14814  14817  14820  14823  14829  14832  14838  14844  14850  14859  14865  14868  14871  14890
14893  14896  14899  14902  14908  14914  14920  14929  14935  14938  14941  14954  14957  14960  14963
14966  14972  14978  14984  14987  14993  15009  15012  15015  15018  15021  15024  15027  15033  15040
15046
```

MULT	176#														
NEWTST	176#														
OR	1#	3323	3421	7553	7610	7814	10051	10119	14540	14719	14728	14752	14826	14841	14911
	14975														
POP	176#	1220	1221	1378	1694	1695	2249								
PUSH	176#	1181	1183	1204	1352	1668	1674	2212							
REPORT	1#	176#													

```
SCOPE    71#  15166
SETBRI  176#
SETTRA 1544#  1555  1556  1557  1558  1559  1560  1561  1562  1563  1564  1565  1566  1567  1568
       1569
SETUP   176#
SHFBRT    1#  2358  2438  2555  2635  2752  2832  2949  3029  3156  3245  3382  3480  3597  3677
       3794  3874  3991  4071  4188  4268  4385  4465  4582  4662  4779  4859  4976  5056  5356
       5441  5563  5648  5770  5855  5977  6062  6184  6269  6391  6476  6598  6683  6805  6890
       7012  7097  7219  7304  7370  7562  7948  8228  8385
SKIP    176#
SLASH   176#
SPACE   176#
STARS   176#   219   242   294   297   483   485   492   961  1028  1097  1176  1235  1241  1270
       1338  1523  1664  1680
SUBWC     1#  11953  12021
SUBI      1#
SUBIC     1#  8541  8625  8701  8791  11319  11387
SUB2C     1#  2305  2385  2502  2582  2699  2779  2896  2976  3103  3192  3329  3427  3544  3624
       3741  3821  3938  4018  4135  4215  4332  4412  4529  4609  4726  4806  4923  5003  5120
       5190  5300  5385  5507  5592  5714  5799  5921  6006  6128  6213  6335  6420  6542  6627
       6749  6834  6956  7041  7163  7248  7418  7455  7655  7695  8053  8175  8246  8332  8406
       8482  8572  8732  9106  9177  9423  9494  9740  9811  10057  10128  10374  10445  10691  10762
      11008  11079  11325  11396  11642  11713  11959  12030  12276  12347  12593  12664  12910  12981  13221
      13227  13289  13298  13544  13615  13979  14360  14425  14612  14631
SWRSU   176#
TRAPTRP 1544#
TYPBIN  176#
TYPDEC  176#
TYPNAM  176#
TYPNUM  176#
TYPOCS  176#
TYPOCT  176#
TYPTXT  176#
SADC      1#  5161  5231  8008  8096  8237  8397  8532  8616  8695  8782  8864  8915  12904  12972
      13791  13967  14769  14772
SADD      1#  9094  9165  9230  9411  9482  9547  9728  9799  9864  10045  10116  10181  10362  10433
      10498  10679  10685  10750  10753  10815  10996  11067  11132  11313  11384  11449  11630  11701  11766
      11947  12018  12083  12264  12335  12400  12581  12652  12717  12898  12969  13034  13215  13285  13351
      13532  13603  13668  14862  14932
SALUTO    30#  8811
SALUT1    30#  8934  9251  9568  9885  10202  10519  10836  11153  11470  11787  12104  12421  12738  13055
      13372
SASL      1#  12587  12655
SAUTO     1#  817
SBEGIN    1#  2281  2478  2675  2872  3070  3287  3520  3717  3914  4111  4308  4505  4702  4899
       5097  5274  5481  5678  5895  6102  6309  6516  6723  6930  7137  7343  7517  7760  7897
       8137  8294  8455  8668  8834  8960  9277  9594  9911  10228  10545  10862  11179  11496  11813
      12130  12447  12764  13081  13398  13713  13966  14460  14698  14805
SBR       1#  2333  2339  2346  2352  2364  2413  2419  2426  2432  2449  2452  2530  2536  2543
       2549  2561  2610  2616  2623  2629  2646  2649  2727  2733  2740  2746  2758  2807  2813
       2820  2826  2843  2849  2924  2930  2937  2943  2955  3004  3010  3017  3023  3040  3043
       3131  3137  3144  3150  3162  3220  3226  3233  3239  3256  3259  3357  3363  3370  3376
       3388  3455  3461  3468  3474  3491  3494  3572  3578  3585  3591  3603  3652  3658  3665
       3671  3688  3691  3769  3775  3782  3788  3800  3849  3855  3862  3868  3885  3888  3966
       3972  3979  3985  3997  4046  4052  4059  4065  4082  4085  4163  4169  4176  4182  4194
       4243  4249  4256  4262  4279  4282  4360  4366  4373  4379  4391  4440  4446  4453  4459
```

```
                4476    4479    4557    4563    4570    4576    4588    4637    4643    4650    4656    4673    4676    4754    4760
                4767    4773    4785    4834    4840    4847    4853    4870    4873    4951    4957    4964    4970    4982    5031
                5037    5044    5050    5067    5070    5139    5142    5149    5152    5170    5209    5212    5219    5222    5245
                5248    5331    5337    5344    5350    5363    5416    5422    5429    5438    5452    5455    5538    5544    5551
                5557    5569    5623    5629    5636    5642    5659    5662    5745    5751    5758    5764    5776    5830    5836
                5843    5849    5866    5869    5952    5958    5965    5971    5983    6037    6043    6050    6056    6073    6076
                6159    6165    6172    6178    6190    6244    6250    6257    6263    6280    6283    6366    6372    6379    6385
                6397    6451    6457    6464    6470    6487    6490    6573    6579    6586    6592    6604    6658    6664    6671
                6677    6694    6697    6780    6786    6793    6799    6811    6865    6871    6878    6884    6901    6904    6987
                6993    7000    7006    7018    7072    7078    7085    7091    7108    7111    7194    7200    7207    7213    7225
                7279    7285    7292    7298    7315    7318    7443    7446    7477    7480    7487    7490    7660    7683    7720
                7723    7730    7733    7835    7846    7851    7867    7870    7957    8011    8078    8081    8099    8106    8109
                8203    8209    8216    8222    8259    8266    8269    8360    8366    8373    8379    8391    8419    8426    8429
                8513    8516    8523    8526    8547    8597    8600    8607    8610    8631    8639    8642    8707    8757    8763
                8770    8776    8797    8805    8808    8864    8867    8894    8947    8928    8931    9134    9140    9147    9153
                9205    9211    9218    9224    9238    9245    9248    9451    9457    9464    9470    9522    9528    9535    9541
                9555    9562    9565    9768    9774    9781    9787    9833    9845    9852    9858    9872    9879    9882    10085
              10091   10098   10104   10156   10162   10169   10175   10189   10196   10199   10402   10408   10415   10421   10473
              10479   10486   10492   10506   10513   10516   10719   10725   10732   10738   10790   10796   10803   10809   10823
              10830   10833   11036   11042   11049   11055   11107   11113   11120   11126   11140   11147   11150   11353   11359
              11366   11372   11424   11430   11437   11443   11457   11464   11467   11670   11676   11683   11689   11741   11747
              11754   11760   11774   11781   11784   11987   11993   12000   12006   12058   12064   12071   12077   12091   12098
              12101   12304   12310   12317   12323   12375   12381   12388   12394   12408   12415   12418   12621   12627   12634
              12640   12692   12698   12705   12711   12725   12732   12735   12938   12944   12951   12957   13009   13015   13022
              13028   13042   13049   13052   13255   13261   13268   13274   13326   13332   13339   13345   13359   13366   13369
              13572   13578   13585   13591   13643   13649   13656   13662   13676   13683   13686   13744   13747   13762   13776
              13779   13800   13814   13817   13824   13927   13965   13982   13998   14017   14023   14039   14055   14065   14075
              14085   14095   14105   14115   14127   14137   14147   14157   14167   14177   14187   14197   14213   14232   14245
              14258   14271   14284   14297   14310   14323   14336   14382   14385   14394   14403   14410   14413   14450   14454
              14585   14600   14618   14656   14659   14666   14669   14763   14778   14786   14874   14944   14999   15043   15049
SBRFLT     9#   5104    5174
SBUFFE     1#   1781
SCOMP      1#
SCYCLE     1#   1852
SDEC       1#   7951    7996    8084    8900    8912    8981    9232    9298    9549    9615    9866    9932   10183   10249
           10500   10566   10817   10883   11134   11200   11451   11517   11768   11634   12085   12151   12402   12468   12719
           12785   13036   13102   13353   13419   13538   13606   13670   14002   14036   14397
SEOP       1#    953
SERTBL     1#    413
SFINI      1#  15311
SFLOT      3#   2280    2368    2485    2565    2682    2762    2879    2959    3077    3166    3294    3392    3527    3607
           3724    3804    3921    4001    4118    4198    4315    4395    4512    4592    4709    4789    4906    4986
SFLT1      4#   2258    2455    2652    2849    3046    3262    3497    3694    3891    4088    4285    4482    4679    4876
SGETPA     1#
SHEADE     1#     42
SHLRSR     3#  14457
SIFEQ      1#   2302    2382    2499    2579    2696    2776    2893    2973    3100    3180    3326    3424    3541    3621
           3738    3818    3935    4015    4132    4212    4329    4409    4526    4606    4723    4803    4920    5000    5117
           5187    5297    5382    5504    5589    5711    5796    5918    6003    6125    6210    6332    6417    6539    6624
           6746    6831    6953    7038    7160    7245    7415    7452    7652    7692    8050    8172    8329    8479    8569
           8729    9103    9174    9420    9491    9737    9808   10054   10125   10371   10442   10688   10759   11005   11076
          11322   11393   11639   11710   11956   12027   12273   12344   12590   12661   12907   12978   13224   13295   13541
          13612   14357   14422   14628
SIFHI      1#   8538    8622    8698    8788
SIFKIS     1#  13976
SIFLO      1#   8243    8403
```

```
$IFLOS      1#
$IFNE       1#  14609
$INC        1#   7829  8002  8005  8090  8093  8234  8394  8529  8613  8692  8779  8861  8909 12270
           12338 13788 13964 14C11 14388 14766 14808 14811 14884 14887
$LAST       1#
$LOC        1#   2281  2478  2675  2872  3070  3287  3520  3717  3914  4111  4308  4505  4702  4899
            5097  5274  5481  5688  5895  6102  6309  6516  6723  6930  7137  7343  7517  7760  7897
            8137  8294  8455  8668  8834  8960  9277  9594  9911 10228 10545 10862 11179 11496 11813
           12130 12447 12764 13081 13398 13713 13856 14480 14698 14807
$LUCLK      29#
$MARHI      1#
$MEM10      21#  8115
$MEM11      19#  8272
$MEM2       22#  8432
$MEM31      23#  8645
$MMFLT      18#  8137  8294
$MOCK       1#
$MSG        1#  1769
$NOISE      30#  13830
$NPRG       28#
$NPR1       25#  7321
$NPR2       26#  7494
$NPR4       27#  7873
$NPR9       28#  7737
$PASEN      1#   967
$PFAIL      1#  1659
$PGMCK      29#  13689
$PROC3      6#
$PROC4      9#  5073
$PWR1       30#  14673
$QUEST      1#  1967  1979  1987  2044  2052
$RAMCL      1#  1713
$RCLK       1#  1716  1719  1756  1761
$ROL        1#  5158  5228 11002 11070
$ROVAR      1#   363
$SCADD      1#  1039
$SCAD1      1#  1078
$SIMBC      1#
$SOFTC      1#  1789
$SPFLT      12#  5281  5366  5488  5573  5695  5780  5902  5987  6109  6194  6316  6401  6523  6608
            6730  6815  6937  7022  7144  7229
$SPTS1      13#  5251  5458  5665  5872  6079  6286  6493  6700  6907  7114
$SP2        14#
$SUBR       1#
$TSTN       1#  2269  2466  2663  2860  3058  3275  3508  3705  3902  4099  4296  4493  4690  4887
            5085  5262  5469  5676  5883  6090  6297  6504  6711  6918  7125  7331  7505  7748  7885
            8125  8282  8443  8656  8822  8948  9265  9582  9899 10216 10533 10850 11167 11484 11801
           12118 12435 12752 13069 13386 13701 13844 14468 14686
$UPADD      1#  1687
$VARIA      1#   235
$XOR        1#  10368 10436
$XZ         1#  2259  2266  2456  2463  2653  2660  2850  2857  3047  3055  3263  3272  3498  3505
            3695  3702  3892  3899  4089  4096  4286  4293  4483  4490  4680  4687  4877  4884  5074
            5082  5252  5259  5459  5466  5666  5673  5873  5880  6080  6087  6287  6294  6494  6501
            6701  6708  6908  6915  7115  7122  7322  7328  7495  7502  7738  7745  7874  7882  8116
            8122  8273  8279  8433  8440  8646  8653  8812  8819  8935  8945  9252  9262  9569  9579
```

```
                9896    9896   10203   10213   10520   10530   10837   10847   11154   11164   11471   11481   11788   11798   12105
               12115   12422   12432   12739   12749   13056   13066   13373   13383   13690   13698   13831   13841   14458   14465
               14674  *14682
$$CMRE          240#     279     280     281     282     283     284
$$CMTM          240#     285     286     287     288     289
$$ESCA          176#
$$MEXT          176#
$$SCOP            1#    1022
$$SET          1544#    1555    1556    1557    1558    1559    1560    1561    1562    1563    1564    1565    1566    1567    1568
               1569
$$SKIP          176#
.BPAGE            1#   14456   14876   14946   15001
.BRCTL            1#    2309    2334    2340    2347    2353    2362    2365    2369    2414    2420    2427    2433    2442    2450
               2457    2506    2531    2537    2544    2550    2559    2563    2606    2611    2617    2624    2630    2639    2647
               2650    2703    2728    2734    2741    2747    2756    2759    2763    2808    2814    2821    2827    2836    2844
               2847    2900    2925    2931    2938    2944    2953    2956    2960    3005    3011    3018    3024    3033    3041
               3044    3107    3132    3138    3145    3151    3160    3163    3167    3221    3227    3234    3240    3249    3257
               3260    3333    3358    3364    3371    3377    3386    3389    3431    3456    3462    3469    3475    3484    3492
               3495    3548    3573    3579    3586    3592    3601    3604    3608    3652    3658    3666    3672    3681    3689
               3692    3745    3770    3776    3783    3789    3798    3801    3805    3850    3856    3863    3869    3878    3886
               3889    3942    3967    3973    3980    3986    3995    3998    4002    4047    4053    4060    4066    4075    4083
               4086    4139    4164    4170    4177    4183    4192    4195    4199    4244    4250    4257    4263    4272    4280
               4283    4336    4361    4367    4374    4380    4389    4392    4396    4441    4447    4454    4460    4469    4477
               4480    4533    4558    4564    4571    4577    4586    4589    4593    4638    4644    4651    4657    4666    4674
               4677    4730    4755    4761    4768    4774    4783    4786    4790    4835    4841    4848    4854    4863    4871
               4874    4927    4952    4958    4965    4971    4980    4983    4987    5032    5038    5045    5051    5060    5068
               5071    5124    5140    5143    5150    5153    5162    5165    5171    5210    5213    5220    5223    5238    5246
               5249    5304    5330    5333    5340    5343    5352    5355    5359    5407    5413    5420    5423    5438    5453
               5456    5511    5536    5539    5546    5549    5558    5561    5565    5604    5610    5617    5621    5636    5660
               5663    5718    5746    5749    5756    5759    5768    5771    5774    5811    5817    5824    5828    5843    5867
               5870    5925    5953    5956    5963    5966    5975    5978    5981    6018    6024    6031    6035    6050    6074
               6077    6132    6160    6163    6170    6173    6182    6185    6189    6225    6231    6238    6242    6257    6281
               6284    6339    6367    6370    6377    6380    6389    6392    6396    6432    6438    6445    6449    6464    6488
               6491    6546    6574    6577    6584    6587    6596    6599    6603    6639    6645    6652    6656    6671    6695
               6698    6753    6781    6784    6791    6794    6803    6806    6810    6846    6852    6859    6863    6878    6902
               6905    6960    6988    6991    6998    7001    7010    7013    7017    7053    7059    7066    7072    7085    7109
               7112    7167    7195    7198    7205    7208    7217    7220    7224    7260    7266    7273    7279    7292    7316
               7319    7410    7432    7435    7442    7445    7454    7457    7461    7497    7503    7510    7516    7529    7553
               7699    7721    7724    7727    7734    7737    7746    7749    7753    7790    7796    7803    7809    7822    7846
               8000    8012    8057    8060    8067    8070    8079    8082    8086    8123    8129    8136    8142    8155    8179
               8250    8262    8267    8270    8277    8280    8289    8292    8296    8333    8339    8346    8352    8365    8389
               8486    8514    8517    8520    8527    8530    8539    8542    8546    8583    8589    8596    8602    8615    8639
               8643    8705    8708    8711    8718    8721    8730    8733    8737    8774    8780    8787    8793    8806    8830
               8833    8898    8922    8925    8932    8935    8944    8947    8951    8988    8994    9001    9007    9020    9044
               9047    9119    9146    9149    9156    9159    9168    9171    9175    9212    9218    9225    9231    9244    9268
               9271    9343    9369    9372    9379    9382    9391    9394    9398    9435    9441    9448    9454    9467    9491
               9494    9567    9593    9596    9603    9606    9615    9618    9622    9659    9665    9672    9678    9691    9715
               9718    9791    9817    9820    9827    9830    9839    9842    9846    9883    9889    9896    9902    9915    9939
               9942    9967    9993    9996   10003   10006   10015   10018   10022   10059   10065   10072   10078   10091   10115
              10118   10191   10217   10220   10227   10230   10239   10242   10246   10283   10289   10296   10302   10315   10339
              10342   10415   10441   10444   10451   10454   10463   10466   10470   10507   10513   10520   10526   10539   10563
              10566
```

```
                13340   13346   13357   13360   13367   13370   13548   13573   13579   13586   13592   13619   13644   13650   13657
                13663   13674   13677   13684   13687   13732   13745   13748   13760   13763   13777   13780   13786   13798   13801
                13815   13818   13825   13828   13903   13946   13993   13999   14006   14015   14018   14024   14040   14056   14062
                14076   14086   14096   14106   14116   14128   14138   14148   14158   14168   14178   14188   14198   14214   14233
                14246   14259   14272   14285   14298   14311   14324   14337   14364   14383   14386   14392   14395   14401   14404
                14411   14414   14429   14451   14465   14547   14559   14571   14577   14586   14592   14601   14616   14619   14635
                14657   14660   14667   14670   14736   14741   14761   14764   14776   14779   14787   14848   14854   14857   14875
                14918   14924   14927   14945   14982   14997   15000   15031   15037   15044   15050
.DEST       18  2283    2286    2290    2294    2297    2300    2306    2312    2315    2318    2321    2324    2327    2331
            2337    2344    2350    2356    2371    2374    2377    2380    2386    2392    2395    2398    2401    2404    2407
            2411    2417    2424    2430    2436    2447    2480    2483    2487    2491    2494    2497    2503    2509    2512
            2515    2518    2521    2524    2528    2534    2541    2547    2553    2567    2571    2574    2577    2583    2589
            2592    2596    2598    2601    2604    2608    2614    2621    2627    2633    2644    2677    2680    2684    2688
            2691    2694    2700    2706    2709    2712    2715    2718    2721    2725    2731    2738    2744    2750    2764
            2768    2771    2774    2780    2786    2789    2792    2796    2798    2801    2805    2811    2818    2824    2830
            2841    2874    2877    2881    2885    2888    2891    2897    2903    2906    2909    2912    2915    2918    2922
            2928    2935    2941    2947    2961    2965    2968    2971    2977    2983    2986    2989    2992    2995    2998
            3002    3008    3015    3021    3027    3038    3072    3075    3079    3083    3086    3089    3092    3095    3098
            3104    3110    3113    3116    3119    3122    3126    3129    3136    3142    3148    3154    3168    3172    3175
            3178    3181    3184    3187    3193    3199    3202    3205    3208    3211    3214    3218    3224    3231    3237
            3243    3254    3289    3292    3296    3300    3303    3306    3309    3312    3315    3318    3321    3324    3330
            3336    3339    3342    3345    3348    3351    3355    3361    3364    3374    3380    3394    3398    3401    3404
            3407    3410    3413    3416    3419    3422    3425    3434    3437    3440    3443    3446    3449    3453    3459
            3466    3472    3478    3489    3502    3505    3508    3533    3536    3539    3545    3551    3554    3557    3560
            3563    3566    3570    3576    3583    3586    3589    3609    3613    3616    3619    3625    3631    3634    3637
            3640    3643    3646    3650    3656    3663    3669    3675    3686    3719    3722    3726    3730    3733    3736
            3742    3748    3751    3754    3757    3760    3763    3767    3773    3780    3786    3792    3806    3810    3813
            3816    3822    3828    3831    3834    3837    3840    3843    3847    3853    3860    3866    3872    3883    3916
            3919    3923    3927    3930    3933    3939    3945    3948    3951    3954    3957    3960    3964    3970    3977
            3983    3989    4003    4007    4010    4013    4019    4025    4028    4031    4034    4037    4040    4044    4050
            4057    4063    4069    4080    4113    4116    4120    4124    4127    4130    4136    4142    4145    4148    4151
            4154    4157    4161    4167    4174    4180    4186    4200    4204    4207    4210    4216    4222    4225    4228
            4231    4234    4237    4241    4247    4254    4260    4266    4277    4310    4313    4317    4321    4324    4327
            4333    4339    4342    4345    4348    4351    4354    4368    4364    4371    4377    4383    4397    4401    4404
            4407    4413    4419    4422    4425    4428    4431    4434    4438    4444    4451    4457    4463    4474    4507
            4510    4514    4518    4521    4524    4530    4536    4539    4542    4545    4548    4551    4555    4561    4568
            4574    4580    4594    4598    4601    4604    4610    4616    4619    4622    4625    4628    4631    4635    4641
            4648    4654    4660    4671    4704    4707    4711    4715    4718    4721    4727    4733    4736    4739    4742
            4745    4748    4752    4758    4765    4771    4777    4791    4796    4798    4801    4807    4813    4816    4819
            4822    4825    4828    4832    4838    4845    4851    4857    4868    4901    4904    4908    4912    4915    4918
            4824    4930    4933    4936    4939    4942    4945    4949    4955    4962    4969    4974    4988    4992    4995
            4998    5004    5010    5013    5016    5019    5022    5025    5029    5035    5042    5048    5054    5065    5099
            5102    5106    5109    5112    5115    5121    5127    5130    5133    5137    5147    5156    5165    5176    5179
            5182    5188    5191    5197    5200    5203    5207    5217    5226    5243    5276    5279    5283    5286    5348
            5289    5292    5295    5301    5307    5310    5313    5316    5319    5325    5328    5335    5342    5345    5348
            5354    5368    5371    5374    5377    5380    5386    5392    5396    5398    5401    5404    5407    5410    5414
            5420    5427    5433    5439    5450    5483    5486    5490    5493    5496    5499    5502    5508    5514    5517
            5520    5523    5526    5529    5532    5536    5542    5549    5565    5561    5575    5578    5581    5584    5587
            5593    5599    5602    5605    5608    5611    5614    5617    5621    5627    5634    5640    5646    5657    5690
            5693    5697    5700    5703    5706    5709    5715    5721    5724    5727    5730    5733    5736    5739    5743
            5749    5756    5762    5768    5782    5785    5788    5791    5794    5800    5806    5809    5812    5815    5818
            5821    5824    5828    5834    5841    5847    5853    5864    5897    5900    5904    5907    5910    5913    5916
            5922    5928    5931    5934    5937    5940    5943    5946    5950    5956    5963    5969    5975    5989    5992
            5995    5998    6001    6007    6013    6016    6019    6022    6025    6028    6031    6035    6041    6048    6054
            6060    6071    6104    6107    6111    6114    6117    6120    6123    6129    6135    6138    6141    6144    6147
            6150    6153    6157    6163    6170    6176    6182    6196    6199    6202    6205    6208    6214    6220    6223
```

6226	6229	6232	6235	6238	6242	6248	6255	6261	6267	6278	6311	6314	6318	6321
6324	6327	6330	6336	6342	6345	6348	6351	6354	6357	6360	6364	6370	6377	6383
6389	6403	6406	6409	6412	6415	6421	6427	6430	6433	6436	6439	6442	6445	6449
6455	6462	6468	6474	6485	6518	6521	6525	6528	6531	6534	6537	6543	6549	6552
6555	6558	6561	6564	6567	6571	6577	6584	6590	6596	6610	6613	6616	6619	6622
6628	6634	6637	6640	6643	6646	6649	6652	6656	6662	6669	6675	6681	6692	6725
6728	6732	6735	6738	6741	6744	6750	6756	6759	6763	6765	6768	6771	6774	6778
6784	6791	6797	6803	6817	6820	6823	6826	6829	6835	6841	6844	6847	6850	6853
6856	6859	6863	6869	6876	6882	6888	6899	6932	6935	6938	6942	6945	6948	6951
6957	6963	6966	6969	6972	6975	6978	6981	6985	6991	6998	7004	7010	7024	7027
7030	7033	7036	7042	7048	7051	7054	7057	7060	7063	7066	7070	7076	7083	7089
7096	7106	7139	7142	7146	7149	7152	7155	7158	7164	7170	7173	7176	7179	7182
7186	7188	7196	7198	7205	7211	7217	7231	7234	7237	7240	7243	7249	7255	7258
7261	7264	7267	7270	7273	7277	7283	7290	7296	7302	7313	7344	7347	7350	7353
7356	7359	7362	7365	7368	7374	7377	7380	7383	7386	7389	7392	7395	7398	7401
7404	7407	7413	7419	7425	7428	7431	7434	7437	7441	7450	7456	7462	7465	7468
7471	7475	7485	7518	7521	7524	7527	7530	7533	7536	7539	7542	7545	7548	7551
7554	7557	7560	7566	7569	7572	7575	7578	7581	7584	7587	7590	7593	7596	7599
7602	7605	7608	7611	7614	7623	7626	7629	7632	7635	7638	7641	7647	7650	7656
7662	7665	7668	7671	7674	7678	7687	7690	7696	7702	7705	7708	7711	7714	7718
7728	7761	7764	7767	7770	7773	7776	7779	7782	7785	7788	7791	7794	7797	7800
7803	7806	7809	7812	7815	7818	7821	7824	7839	7842	7846	7855	7858	7861	7865
7898	7901	7904	7907	7910	7913	7916	7919	7922	7925	7928	7931	7934	7937	7940
7943	7946	7961	7964	7967	7970	7973	7976	7979	7982	7985	7991	7994	8015	8018
8021	8024	8027	8030	8033	8036	8039	8042	8045	8048	8054	8060	8063	8066	8069
8072	8076	8104	8139	8142	8145	8148	8151	8154	8157	8160	8164	8167	8170	8176
8182	8185	8188	8191	8194	8197	8201	8207	8214	8220	8226	8241	8247	8254	8257
8264	8296	8299	8302	8305	8308	8311	8314	8317	8321	8324	8327	8333	8339	8342
8345	8348	8351	8354	8358	8364	8371	8377	8383	8401	8407	8414	8417	8424	8456
8459	8462	8465	8468	8471	8474	8477	8483	8489	8492	8495	8498	8501	8504	8507
8511	8521	8536	8542	8552	8555	8558	8561	8564	8567	8573	8579	8582	8585	8588
8591	8595	8605	8620	8623	8627	8669	8672	8675	8678	8681	8684	8687	8690	8702
8712	8715	8718	8721	8724	8727	8733	8739	8742	8745	8748	8751	8755	8761	8768
8774	8786	8792	8803	8835	8838	8841	8844	8847	8850	8853	8856	8859	8869	8875
8878	8882	8892	8904	8907	8919	8926	8961	8964	8967	8970	8973	8976	8979	8985
8988	8991	8994	8997	9000	9003	9006	9009	9012	9015	9016	9021	9024	9027	9030
9033	9036	9039	9042	9045	9048	9051	9054	9057	9060	9063	9066	9069	9072	9075
9078	9081	9084	9088	9092	9098	9101	9107	9113	9116	9119	9122	9125	9128	9132
9138	9145	9151	9157	9160	9163	9169	9172	9178	9184	9187	9190	9193	9196	9199
9203	9209	9216	9222	9228	9243	9270	9281	9284	9287	9290	9243	9246	9302	9305
9308	9311	9314	9317	9320	9323	9326	9329	9332	9335	9338	9341	9344	9347	9350
9353	9356	9359	9362	9365	9368	9371	9374	9377	9380	9383	9386	9389	9392	9395
9398	9401	9405	9409	9415	9418	9424	9430	9433	9436	9439	9442	9445	9449	9455
9462	9468	9474	9477	9480	9486	9489	9495	9501	9504	9507	9510	9513	9516	9520
9526	9533	9539	9545	9560	9565	9580	9601	9604	9607	9610	9613	9619	9622	9625
9628	9631	9634	9637	9640	9643	9646	9649	9652	9655	9658	9661	9664	9667	9670
9673	9676	9679	9682	9685	9688	9691	9694	9697	9700	9703	9706	9709	9712	9715
9718	9722	9726	9732	9736	9741	9747	9750	9753	9756	9759	9762	9766	9772	9779
9785	9791	9794	9797	9803	9806	9812	9818	9821	9824	9827	9830	9833	9837	9843
9850	9856	9862	9877	9912	9915	9918	9921	9924	9927	9930	9936	9939	9942	9945
9948	9951	9954	9957	9960	9963	9966	9969	9972	9975	9978	9981	9984	9987	9990
9993	9996	9999	10002	10005	10008	10011	10014	10017	10020	10023	10026	10029	10032	10035
10039	10043	10049	10052	10058	10064	10067	10070	10073	10076	10079	10083	10089	10096	10102
10108	10111	10114	10120	10123	10129	10135	10138	10141	10144	10147	10150	10154	10160	10167
10173	10179	10194	10229	10232	10235	10238	10241	10244	10247	10253	10256	10259	10262	10265
10268	10271	10274	10277	10280	10283	10286	10289	10292	10295	10298	10301	10304	10307	10310

```
10313  10316  10319  10322  10325  10328  10331  10334  10337  10340  10343  10346  10349  10352  10356
10360  10366  10369  10375  10381  10384  10387  10390  10393  10396  10400  10406  10413  10419  10425
10428  10431  10437  10440  10446  10452  10455  10458  10461  10464  10467  10471  10477  10484  10490
10496  10511  10546  10549  10552  10555  10558  10561  10564  10570  10573  10576  10579  10582  10585
10588  10591  10594  10597  10600  10603  10606  10609  10612  10615  10618  10621  10624  10627  10630
10633  10636  10639  10642  10645  10648  10651  10654  10657  10660  10663  10666  10669  10673  10677
10683  10686  10692  10698  10701  10704  10707  10710  10713  10717  10723  10730  10736  10742  10745
10748  10754  10757  10763  10769  10772  10775  10778  10781  10784  10788  10794  10801  10807  10813
10820  10823  10826  10829  10832  10835  10838  10841  10847  10850  10853  10856  10859  10902  10905
10908  10911  10914  10917  10920  10923  10926  10929  10932  10936  10938  10941  10944  10947  10950
10953  10956  10959  10962  10965  10968  10971  10974  10977  10980  10983  10986  10990  10994  11000
11003  11009  11015  11018  11021  11024  11027  11030  11034  11040  11047  11053  11059  11062  11065
11071  11074  11080  11086  11089  11092  11095  11098  11101  11105  11111  11118  11124  11130  11145
11180  11183  11186  11189  11192  11195  11198  11204  11207  11210  11213  11216  11219  11222  11225
11228  11231  11234  11237  11240  11243  11246  11249  11252  11255  11258  11261  11264  11267  11270
11273  11276  11279  11282  11285  11288  11291  11294  11297  11300  11303  11307  11311  11317  11320
11326  11332  11335  11338  11341  11344  11347  11351  11357  11364  11370  11376  11379  11382  11388
11391  11397  11403  11406  11409  11412  11415  11418  11422  11428  11435  11441  11447  11462  11497
11500  11503  11506  11509  11512  11515  11521  11524  11527  11530  11533  11536  11539  11542  11545
11548  11551  11554  11557  11560  11563  11566  11569  11572  11575  11578  11581  11584  11587  11590
11593  11596  11599  11602  11605  11608  11611  11614  11617  11620  11624  11628  11634  11637  11643
11649  11652  11655  11658  11661  11664  11668  11674  11681  11687  11693  11696  11699  11705  11708
11714  11720  11723  11726  11729  11732  11736  11739  11745  11752  11758  11764  11773  11814  11817
11820  11823  11826  11829  11832  11838  11841  11844  11847  11850  11853  11856  11859  11862  11865
11868  11871  11874  11877  11880  11883  11886  11889  11892  11895  11898  11901  11904  11907  11910
11913  11916  11919  11922  11925  11928  11931  11934  11937  11941  11945  11951  11954  11958  11966
11969  11972  11975  11978  11981  11988  11991  11998  12004  12010  12013  12016  12022  12025  12031
12037  12040  12043  12046  12049  12052  12056  12062  12069  12075  12081  12096  12131  12134  12137
12140  12143  12146  12149  12155  12158  12161  12164  12167  12170  12173  12176  12179  12182  12185
12188  12191  12194  12197  12200  12203  12206  12209  12212  12215  12218  12221  12224  12227  12230
12233  12236  12239  12242  12245  12248  12251  12254  12257  12260  12263  12274  12277  12283  12286
12289  12292  12295  12298  12302  12308  12315  12321  12327  12330  12333  12339  12342  12348  12354
12357  12360  12363  12366  12369  12373  12379  12382  12385  12388  12391  12413  12418  12451  12457
12460  12463  12466  12472  12475  12478  12481  12484  12487  12490  12493  12496  12499  12502  12505
12508  12511  12514  12517  12520  12523  12526  12529  12532  12535  12538  12541  12544  12547  12550
12553  12556  12559  12562  12565  12568  12571  12575  12578  12581  12586  12594  12600  12603  12606
12609  12612  12615  12619  12625  12630  12638  12644  12647  12650  12656  12659  12665  12671  12674
12677  12680  12683  12686  12689  12692  12696  12703  12709  12730  12765  12768  12771  12774  12777
12780  12783  12789  12792  12795  12798  12801  12804  12807  12810  12813  12816  12819  12822  12825
12828  12831  12834  12837  12840  12843  12846  12849  12852  12855  12858  12861  12864  12867  12870
12873  12876  12879  12882  12886  12889  12892  12896  12902  12905  12908  12911  12917  12920  12926
12929  12932  12936  12942  12946  12949  12952  12955  12958  12961  12967  12973  12976  12988  12991  12994
12997  13000  13003  13007  13013  13020  13026  13032  13047  13082  13085  13088  13091  13094  13097
13100  13106  13109  13112  13115  13118  13121  13124  13127  13130  13133  13136  13139  13142  13145
13148  13151  13154  13157  13160  13163  13166  13169  13172  13175  13178  13181  13184  13187  13190
13193  13196  13199  13202  13205  13209  13213  13219  13222  13228  13234  13237  13240  13243  13246
13249  13253  13259  13266  13272  13278  13281  13284  13290  13293  13299  13305  13308  13311  13314
13317  13320  13324  13330  13337  13343  13349  13364  13399  13402  13405  13408  13411  13414  13417
13423  13426  13429  13432  13435  13438  13441  13444  13447  13450  13453  13456  13459  13462  13465
13468  13471  13474  13477  13480  13483  13486  13489  13492  13495  13498  13501  13504  13507  13510
13513  13516  13519  13522  13526  13530  13536  13539  13545  13551  13554  13557  13560  13563  13566
13570  13576  13583  13589  13595  13598  13601  13607  13610  13616  13620  13626  13629  13631  13634
13637  13641  13647  13654  13660  13666  13681  13714  13717  13720  13805  13808  13812  13822  13857  13860  13738
13742  13751  13754  13757  13767  13770  13774  13783  13795  13798  13892  13895  13900  13903  13906  13909
13863  13866  13870  13873  13879  13879  13882  13885  13888  13892  13895  13900  13903  13906  13909
13912  13915  13918  13921  13927  13930  13933  13936  13939  13942  13945  13948  13951  13954  13957
```

```
                13962   13971   13974   13980   13990   13996   14009   14021   14027   14031   14034   14046   14049   14053   14059
                14063   14069   14073   14079   14083   14089   14093   14099   14103   14109   14113   14121   14125   14131   14135
                14141   14145   14151   14155   14161   14165   14171   14175   14181   14185   14191   14195   14204   14207   14211
                14217   14220   14223   14226   14230   14236   14239   14243   14249   14252   14256   14262   14265   14269   14275
                14278   14282   14288   14291   14295   14301   14304   14308   14314   14317   14321   14327   14330   14334   14343
                14346   14349   14352   14355   14361   14367   14370   14373   14376   14380   14408   14417   14420   14426   14432
                14435   14438   14441   14444   14448   14481   14484   14497   14490   14493   14496   14499   14502   14505   14508
                14511   14514   14517   14520   14523   14526   14529   14532   14535   14538   14541   14544   14550   14553   14556
                14562   14565   14568   14574   14580   14583   14509   14595   14598   14604   14607   14613   14623   14626   14632
                14638   14641   14644   14647   14650   14654   14664   14699   14702   14705   14708   14711   14714   14717   14720
                14723   14726   14729   14732   14738   14747   14750   14753   14758   14784   14815   14818   14821   14824   14827
                14830   14833   14836   14839   14842   14845   14851   14860   14863   14866   14869   14872   14891   14894   14897
                14900   14903   14906   14909   14912   14915   14921   14930   14933   14936   14939   14942   14955   14958   14961
                14964   14967   14970   14973   14976   14979   14985   14988   14991   14994   15010   15013   15016   15019   15022
                15025   15028   15034   15041   15047
.DESTM      18  2283    2286    2290    2294    2297    2300    2312    2315    2318    2321    2324    2327    2331    2337
            2344    2350    2356    2370    2374    2377    2380    2392    2395    2398    2401    2404    2407    2411    2417
            2424    2430    2436    2447    2480    2483    2487    2491    2494    2497    2509    2512    2515    2518    2521
            2524    2528    2534    2541    2547    2553    2567    2571    2574    2577    2589    2592    2595    2598    2601
            2604    2608    2614    2621    2627    2633    2644    2677    2680    2684    2688    2691    2694    2706    2709
            2712    2715    2718    2721    2725    2731    2738    2744    2750    2764    2768    2771    2774    2786    2789
            2792    2795    2798    2801    2805    2811    2818    2824    2830    2841    2874    2877    2881    2885    2888
            2891    2903    2906    2909    2912    2915    2918    2922    2928    2935    2941    2947    2961    2965    2968
            2971    2983    2986    2989    2992    2995    2998    3002    3008    3015    3021    3027    3038    3072    3075
            3079    3083    3086    3089    3095    3098    3110    3113    3116    3119    3122    3125    3129    3135    3142
            3148    3154    3168    3172    3175    3178    3184    3187    3199    3202    3205    3208    3211    3214    3218
            3224    3231    3237    3243    3254    3289    3292    3296    3300    3303    3306    3312    3315    3318    3321
            3336    3339    3342    3345    3348    3351    3355    3361    3368    3374    3380    3394    3398    3401    3404
            3410    3413    3416    3419    3434    3437    3440    3443    3446    3449    3453    3459    3466    3472    3478
            3489    3522    3525    3529    3533    3536    3539    3551    3554    3557    3560    3563    3566    3570    3576
            3583    3589    3595    3609    3613    3616    3619    3631    3634    3637    3640    3643    3646    3650    3656
            3663    3669    3675    3686    3719    3722    3726    3730    3733    3736    3748    3751    3754    3757    3760
            3763    3767    3773    3780    3786    3789    3810    3813    3816    3828    3831    3834    3837    3840
            3843    3847    3853    3860    3866    3872    3876    3883    3919    3923    3927    3930    3933    3945    3948
            3951    3954    3957    3960    3964    3970    3977    3989    4003    4007    4010    4013    4025    4028
            4031    4034    4037    4040    4044    4050    4057    4069    4080    4113    4116    4120    4124    4127
            4130    4142    4145    4148    4151    4154    4157    4161    4167    4174    4180    4186    4200    4204    4207
            4210    4222    4225    4228    4231    4234    4237    4241    4247    4254    4260    4266    4277    4310    4313
            4317    4321    4324    4327    4339    4342    4345    4348    4351    4354    4358    4364    4371    4377    4383
            4397    4401    4404    4407    4419    4422    4425    4428    4431    4434    4438    4444    4451    4457    4463
            4474    4507    4510    4514    4518    4521    4524    4536    4539    4542    4545    4548    4551    4555    4561
            4568    4574    4580    4594    4598    4601    4604    4616    4619    4622    4625    4628    4631    4635    4641
            4648    4654    4660    4671    4704    4707    4711    4715    4718    4721    4733    4736    4738    4742    4745
            4748    4752    4758    4765    4771    4777    4781    4795    4798    4801    4813    4816    4819    4822    4825
            4828    4832    4838    4845    4851    4857    4868    4901    4904    4908    4912    4915    4918    4930    4933
            4936    4939    4942    4945    4949    4955    4962    4968    4974    4988    4992    4995    4998    5010    5013
            5016    5019    5022    5025    5029    5035    5042    5048    5054    5065    5099    5102    5106    5109    5112
            5115    5127    5130    5133    5137    5147    5156    5165    5176    5179    5182    5185    5197    5200    5203
            5207    5217    5226    5235    5243    5276    5279    5283    5286    5289    5292    5295    5307    5310    5313
            5316    5319    5322    5325    5329    5335    5342    5348    5354    5360    5371    5374    5377    5380    5382
            5395    5398    5401    5404    5407    5410    5414    5420    5427    5433    5439    5450    5483    5486    5490
            5493    5496    5499    5502    5514    5517    5520    5523    5526    5529    5532    5536    5542    5549    5555
            5561    5575    5578    5581    5584    5587    5599    5602    5605    5608    5611    5614    5617    5621    5627
            5634    5640    5646    5657    5690    5693    5697    5700    5703    5706    5709    5721    5724    5727    5730
            5733    5736    5739    5743    5749    5756    5762    5768    5782    5785    5788    5791    5794    5806    5809
            5812    5815    5818    5821    5824    5828    5834    5841    5847    5853    5864    5897    5900    5904    5907
```

5910	5913	5916	5928	5931	5934	5937	5940	5943	5946	5950	5956	5963	5969	5975
5989	5992	5995	5998	6001	6013	6016	6019	6022	6025	6028	6031	6035	6041	6048
6054	6060	6071	6104	6107	6111	6114	6117	6120	6123	6135	6138	6141	6144	6147
6150	6153	6157	6163	6170	6176	6182	6196	6199	6202	6205	6208	6220	6223	6226
6229	6232	6235	6238	6242	6248	6255	6261	6267	6278	6311	6314	6318	6321	6324
6327	6330	6342	6345	6348	6351	6354	6357	6360	6364	6370	6377	6383	6389	6403
6406	6409	6412	6415	6427	6430	6433	6436	6439	6442	6445	6449	6455	6462	6468
6474	6485	6518	6521	6525	6528	6531	6534	6537	6549	6552	6555	6558	6561	6564
6567	6571	6577	6584	6590	6596	6610	6613	6616	6619	6622	6634	6637	6640	6643
6646	6649	6652	6656	6662	6669	6675	6681	6692	6725	6728	6732	6735	6738	6741
6744	6756	6759	6762	6765	6768	6771	6774	6778	6784	6791	6797	6803	6817	6820
6823	6826	6829	6841	6844	6847	6850	6853	6856	6859	6863	6869	6876	6882	6888
6899	6932	6935	6939	6942	6945	6948	6951	6963	6966	6969	6972	6975	6978	6981
6985	6991	6998	7004	7010	7024	7027	7030	7033	7036	7048	7051	7054	7057	7060
7063	7066	7070	7076	7083	7089	7095	7106	7139	7142	7146	7149	7152	7155	7158
7170	7173	7176	7179	7182	7185	7188	7192	7198	7205	7211	7217	7231	7234	7237
7240	7243	7255	7258	7261	7264	7267	7270	7273	7277	7283	7290	7296	7302	7313
7344	7347	7350	7353	7356	7359	7362	7365	7368	7374	7377	7380	7383	7386	7389
7392	7395	7398	7401	7404	7407	7413	7425	7428	7431	7434	7437	7441	7450	7462
7465	7468	7471	7475	7485	7518	7521	7524	7527	7530	7533	7536	7539	7542	7545
7548	7551	7557	7560	7566	7569	7572	7575	7578	7581	7584	7587	7590	7593	7596
7599	7602	7605	7608	7614	7623	7626	7629	7632	7635	7638	7641	7647	7650	7662
7665	7668	7671	7674	7678	7687	7690	7702	7705	7708	7711	7714	7718	7728	7761
7764	7767	7770	7773	7776	7779	7782	7785	7788	7791	7794	7797	7800	7803	7806
7809	7812	7818	7821	7824	7839	7842	7846	7855	7858	7865	7898	7901	7904	7907
7910	7913	7916	7919	7922	7925	7928	7931	7934	7937	7940	7943	7946	7961	7964
7967	7970	7973	7976	7979	7982	7985	7991	7994	8015	8018	8021	8024	8027	8030
8033	8036	8039	8042	8045	8048	8060	8063	8066	8069	8072	8076	8104	8139	8142
8145	8148	8151	8154	8157	8160	8164	8167	8170	8182	8185	8188	8191	8194	8197
8201	8207	8214	8220	8226	8241	8254	8257	8264	8296	8299	8302	8305	8308	8311
8314	8317	8321	8324	8327	8339	8342	8345	8348	8351	8354	8358	8364	8371	8377
8383	8401	8414	8417	8424	8456	8459	8462	8465	8468	8471	8474	8477	8489	8492
8495	8498	8501	8504	8507	8511	8521	8536	8552	8555	8558	8561	8564	8567	8579
8582	8585	8588	8591	8595	8605	8620	8637	8669	8672	8675	8678	8681	8684	8687
8690	8712	8715	8718	8721	8724	8727	8739	8742	8745	8748	8751	8755	8761	8768
8774	8786	8803	8835	8838	8841	8844	8847	8850	8853	8856	8859	8868	8875	8878
8882	8892	8904	8907	8919	8926	8961	8964	8967	8970	8973	8976	8979	8985	8988
8991	8994	8997	9000	9003	9006	9009	9012	9015	9018	9021	9024	9027	9030	9033
9036	9039	9042	9045	9048	9051	9054	9057	9060	9063	9066	9069	9072	9075	9078
9081	9084	9088	9092	9098	9101	9113	9116	9119	9122	9125	9128	9132	9138	9145
9151	9157	9160	9163	9169	9172	9184	9187	9190	9193	9196	9199	9203	9209	9216
9222	9228	9243	9278	9281	9284	9287	9290	9293	9296	9302	9305	9308	9311	9314
9317	9320	9323	9326	9329	9332	9335	9338	9341	9344	9347	9350	9353	9356	9359
9362	9365	9368	9371	9374	9377	9380	9383	9386	9389	9392	9395	9398	9401	9405
9409	9415	9430	9433	9436	9439	9442	9445	9449	9455	9462	9468	9474	9477	9480
9489	9501	9504	9507	9510	9513	9516	9520	9526	9533	9539	9545	9560	9595	9598
9601	9604	9607	9610	9613	9619	9622	9625	9628	9631	9634	9637	9640	9643	9646
9649	9652	9655	9658	9661	9664	9667	9670	9673	9676	9679	9682	9685	9688	9691
9694	9697	9700	9703	9706	9709	9712	9715	9718	9722	9726	9732	9747	9750	9753
9756	9759	9762	9766	9772	9779	9785	9791	9794	9797	9806	9818	9821	9824	9827
9830	9833	9837	9843	9850	9856	9862	9877	9912	9915	9918	9921	9924	9927	9530
9936	9939	9942	9945	9948	9951	9954	9957	9960	9963	9966	9969	9972	9975	9978
9981	9984	9987	9990	9993	9996	9999	10002	10005	10008	10011	10014	10017	10020	10023
10026	10029	10032	10035	10039	10043	10049	10064	10067	10070	10073	10076	10079	10083	10089
10096	10102	10108	10111	10114	10123	10135	10138	10141	10144	10147	10150	10154	10160	10167
10173	10179	10194	10229	10232	10235	10238	10241	10244	10247	10253	10256	10259	10262	10265

10268	10271	10274	10277	10280	10283	10286	10289	10292	10295	10298	10301	10304	10307	10310
10313	10316	10319	10322	10325	10328	10331	10334	10337	10340	10343	10346	10349	10352	10356
10360	10366	10381	10384	10387	10390	10393	10396	10400	10406	10413	10419	10425	10428	10431
10440	10452	10455	10458	10461	10464	10467	10471	10477	10484	10487	10492	10511	10546	10549
10552	10555	10558	10561	10564	10570	10573	10576	10579	10582	10585	10588	10591	10594	10597
10600	10603	10606	10609	10612	10615	10618	10621	10624	10627	10630	10633	10636	10639	10642
10645	10648	10651	10654	10657	10660	10663	10666	10669	10673	10677	10683	10698	10701	10704
10707	10710	10713	10717	10723	10730	10736	10742	10745	10748	10757	10769	10772	10775	10778
10781	10784	10788	10794	10801	10807	10813	10820	10863	10866	10869	10872	10875	10878	10881
10887	10890	10893	10896	10899	10902	10905	10908	10911	10914	10917	10920	10923	10926	10929
10932	10935	10938	10941	10944	10947	10950	10953	10956	10959	10962	10965	10968	10971	10974
10977	10980	10983	10986	10990	10994	11000	11015	11018	11021	11024	11027	11030	11034	11040
11047	11053	11059	11062	11065	11074	11086	11089	11092	11095	11098	11101	11105	11111	11118
11124	11130	11145	11180	11183	11186	11189	11192	11195	11198	11204	11207	11210	11213	11216
11219	11222	11225	11228	11231	11234	11237	11243	11243	11246	11249	11252	11255	11258	11261
11264	11267	11270	11273	11276	11279	11282	11285	11288	11291	11294	11297	11300	11303	11307
11311	11317	11332	11336	11338	11341	11344	11347	11351	11357	11364	11370	11376	11379	11382
11391	11403	11406	11409	11412	11415	11418	11422	11428	11435	11441	11447	11462	11497	11500
11503	11506	11509	11512	11515	11521	11524	11527	11530	11533	11536	11539	11542	11545	11548
11551	11554	11557	11560	11563	11566	11569	11572	11575	11578	11581	11584	11587	11590	11593
11596	11599	11602	11605	11608	11611	11614	11617	11620	11624	11628	11634	11649	11652	11655
11658	11661	11664	11668	11674	11681	11687	11693	11696	11699	11708	11720	11723	11726	11729
11732	11735	11739	11745	11752	11758	11764	11779	11814	11817	11820	11823	11826	11829	11832
11838	11841	11844	11847	11850	11853	11856	11859	11862	11865	11868	11871	11874	11877	11880
11883	11886	11889	11892	11895	11898	11901	11904	11907	11910	11913	11916	11919	11922	11925
11928	11931	11934	11937	11941	11945	11951	11966	11969	11972	11975	11978	11981	11985	11991
11998	12004	12010	12013	12016	12025	12037	12040	12043	12046	12049	12052	12056	12062	12069
12075	12081	12086	12131	12134	12137	12140	12143	12146	12149	12155	12158	12161	12164	12167
12170	12173	12176	12179	12182	12185	12188	12191	12194	12197	12200	12203	12206	12209	12212
12215	12218	12221	12224	12227	12230	12233	12236	12239	12242	12245	12248	12251	12254	12258
12262	12268	12283	12286	12289	12292	12295	12298	12302	12308	12315	12321	12327	12330	12333
12342	12354	12357	12360	12363	12366	12369	12373	12379	12386	12392	12398	12413	12448	12451
12454	12457	12460	12463	12466	12472	12475	12478	12481	12484	12487	12490	12493	12496	12499
12502	12505	12508	12511	12514	12517	12520	12523	12526	12529	12532	12535	12538	12541	12544
12547	12550	12553	12556	12559	12562	12565	12568	12571	12575	12579	12585	12600	12603	12606
12609	12612	12615	12619	12622	12625	12632	12638	12644	12647	12650	12659	12671	12674	12677
12680	12683	12686	12690	12696	12699	12703	12709	12715	12720	12765	12768	12771	12774	12777
12780	12783	12789	12792	12795	12798	12801	12804	12807	12810	12813	12816	12819	12822	12825
12828	12831	12834	12837	12840	12843	12846	12849	12852	12855	12858	12861	12864	12867	12870
12873	12876	12879	12882	12885	12888	12892	12896	12902	12917	12920	12923	12926	12929	12932
12936	12942	12949	12955	12961	12964	12967	12976	12988	12991	12994	12997	13000	13003	13007
13013	13020	13026	13032	13047	13082	13085	13088	13091	13094	13047	13100	13106	13109	13112
13115	13118	13121	13124	13127	13130	13133	13136	13139	13142	13145	13148	13151	13154	13157
13160	13163	13166	13169	13172	13175	13178	13181	13184	13187	13190	13193	13196	13199	13202
13205	13209	13213	13219	13234	13237	13240	13243	13246	13249	13253	13259	13266	13272	13278
13281	13284	13293	13305	13308	13311	13314	13317	13320	13324	13330	13337	13343	13349	13364
13399	13402	13405	13408	13411	13414	13417	13423	13426	13429	13432	13435	13438	13441	13444
13447	13450	13453	13456	13459	13462	13465	13468	13471	13474	13477	13480	13483	13486	13489
13492	13495	13498	13501	13504	13507	13510	13513	13516	13519	13522	13526	13530	13536	13551
13554	13557	13560	13563	13566	13570	13576	13583	13589	13595	13598	13601	13610	13622	13625
13628	13631	13634	13637	13641	13647	13654	13660	13666	13681	13714	13717	13720	13723	13726
13729	13735	13738	13742	13751	13754	13757	13767	13770	13774	13783	13795	13805	13808	13812
13822	13857	13860	13863	13866	13870	13873	13876	13879	13882	13885	13888	13892	13895	13900
13903	13906	13909	13912	13915	13918	13921	13927	13930	13933	13936	13939	13942	13945	13948
13951	13954	13957	13962	13971	13974	13990	13996	14009	14021	14027	14031	14034	14046	14049
14053	14059	14063	14069	14073	14079	14083	14089	14093	14099	14103	14109	14113	14121	14125
14131	14135													

14141 14145 14151 14155 14161 14165 14171 14175 14181 14185 14191 14195 14204 14207 14211
14217 14220 14223 14226 14230 14236 14239 14243 14249 14252 14256 14262 14265 14269 14275
14278 14282 14288 14291 14295 14301 14304 14308 14314 14317 14321 14327 14330 14334 14343
14346 14349 14352 14355 14367 14370 14373 14376 14380 14408 14417 14420 14432 14435 14438
14441 14444 14448 14481 14484 14487 14490 14493 14496 14499 14502 14505 14508 14511 14514
14517 14520 14523 14526 14529 14532 14538 14544 14550 14553 14556 14562 14565 14568 14574
14580 14583 14589 14595 14598 14604 14607 14623 14626 14638 14641 14644 14647 14650 14654
14664 14699 14702 14705 14708 14711 14714 14717 14723 14726 14732 14738 14747 14750 14758
14784 14815 14818 14821 14824 14830 14833 14839 14845 14851 14860 14866 14869 14872 14891
14894 14897 14900 14903 14909 14915 14921 14930 14936 14939 14942 14955 14958 14961 14964
14967 14973 14979 14985 14988 14994 15010 15013 15016 15019 15022 15025 15028 15034 15041
15047

.EQUAT 1# 66
.HEADE 1# 31
.MCTL 1# 2283 2286 2290 2294 2297 2300 2306 2312 2315 2318 2321 2324 2327 2331
2337 2344 2350 2356 2370 2374 2377 2380 2386 2392 2395 2398 2401 2404 2407
2411 2417 2424 2430 2436 2447 2480 2483 2487 2491 2494 2497 2503 2509 2512
2515 2518 2521 2524 2528 2534 2541 2547 2553 2567 2571 2574 2577 2583 2589
2592 2595 2598 2601 2604 2608 2614 2621 2627 2633 2644 2677 2683 2684 2688
2691 2694 2700 2706 2709 2712 2715 2718 2721 2725 2731 2738 2744 2750 2764
2768 2771 2774 2780 2786 2789 2792 2795 2798 2801 2805 2811 2818 2824 2830
2841 2874 2877 2881 2885 2888 2891 2897 2903 2906 2909 2912 2915 2918 2922
2928 2935 2941 2947 2961 2965 2968 2971 2977 2983 2986 2989 2992 2995 2998
3002 3008 3015 3021 3027 3038 3072 3075 3079 3083 3086 3089 3092 3095 3098
3104 3110 3113 3116 3119 3122 3125 3129 3135 3142 3148 3154 3168 3172 3175
3178 3181 3184 3187 3193 3199 3202 3205 3208 3211 3214 3218 3224 3231 3237
3243 3254 3289 3292 3296 3300 3303 3306 3309 3312 3315 3318 3321 3324 3330
3336 3339 3342 3345 3348 3351 3355 3361 3368 3374 3380 3394 3398 3401 3404
3407 3410 3413 3416 3419 3422 3426 3434 3437 3440 3443 3446 3449 3453 3459
3466 3472 3478 3482 3486 3489 3492 3493 3496 3539 3545 3551 3554 3557 3560
3563 3566 3570 3576 3583 3586 3589 3609 3613 3616 3619 3625 3631 3634 3637
3640 3643 3646 3650 3656 3663 3666 3675 3679 3719 3722 3726 3730 3733 3736
3742 3748 3751 3754 3757 3760 3763 3767 3773 3780 3786 3792 3806 3810 3813
3816 3822 3828 3831 3834 3837 3840 3843 3847 3853 3860 3866 3872 3883 3916
3919 3923 3927 3930 3933 3939 3945 3948 3951 3954 3957 3960 3964 3970 3977
3983 3989 4003 4007 4010 4013 4019 4025 4028 4031 4034 4037 4040 4044 4050
4057 4063 4069 4080 4113 4116 4120 4124 4127 4130 4136 4142 4145 4148 4151
4154 4157 4161 4167 4174 4180 4186 4200 4204 4207 4210 4216 4222 4225 4228
4231 4234 4237 4241 4247 4254 4260 4266 4277 4310 4313 4317 4321 4324 4327
4333 4339 4342 4345 4348 4351 4354 4358 4364 4371 4377 4383 4397 4401 4404
4407 4413 4419 4422 4425 4428 4431 4434 4438 4444 4451 4457 4463 4474 4507
4510 4514 4518 4521 4524 4530 4536 4539 4542 4545 4548 4551 4555 4561 4568
4574 4580 4594 4598 4601 4604 4610 4616 4619 4622 4625 4628 4631 4635 4641
4648 4654 4660 4671 4704 4707 4711 4715 4718 4721 4727 4733 4736 4739 4742
4745 4748 4752 4758 4765 4771 4777 4791 4795 4798 4801 4807 4813 4816 4819
4822 4825 4828 4832 4838 4845 4851 4857 4868 4901 4904 4908 4912 4915 4918
4924 4930 4933 4936 4939 4942 4945 4949 4955 4962 4968 4974 4988 4992 4995
4998 5004 5010 5013 5016 5019 5022 5025 5029 5035 5042 5048 5054 5065 5099
5102 5106 5109 5112 5115 5121 5127 5130 5133 5137 5147 5156 5165 5176 5179
5182 5185 5191 5197 5200 5203 5207 5217 5226 5235 5243 5276 5279 5283 5286
5289 5292 5295 5301 5307 5310 5313 5316 5319 5322 5325 5329 5335 5342 5348
5354 5368 5371 5374 5377 5380 5386 5392 5395 5398 5401 5404 5407 5410 5414
5420 5427 5433 5439 5450 5483 5486 5490 5493 5496 5499 5502 5514 5517
5520 5523 5526 5529 5532 5536 5542 5549 5555 5561 5575 5578 5581 5584 5587
5593 5599 5602 5605 5608 5611 5614 5617 5621 5627 5634 5640 5646 5657 5690
5693 5697 5700 5703 5706 5709 5715 5721 5724 5727 5730 5733 5736 5739 5743

5749	5756	5762	5768	5782	5785	5788	5791	5794	5800	5806	5809	5812	5815	5818
5821	5824	5828	5834	5841	5847	5853	5864	5897	5900	5904	5907	5910	5913	5916
5922	5928	5931	5934	5937	5940	5943	5946	5950	5956	5963	5969	5975	5989	5992
5995	5998	6001	6007	6013	6016	6019	6022	6025	6028	6031	6035	6041	6048	6054
6060	6071	6104	6107	6111	6114	6117	6120	6123	6129	6135	6138	6141	6144	6147
6150	6153	6157	6163	6170	6176	6182	6196	6199	6202	6205	6208	6214	6220	6223
6226	6229	6232	6235	6238	6242	6248	6255	6261	6267	6278	6311	6314	6318	6321
6324	6327	6330	6336	6342	6345	6348	6351	6354	6357	6360	6364	6370	6377	6383
6389	6403	6406	6409	6412	6415	6421	6427	6430	6433	6436	6439	6442	6445	6448
6455	6462	6468	6474	6485	6518	6521	6525	6528	6531	6534	6537	6543	6549	6552
6555	6558	6561	6564	6567	6571	6577	6584	6590	6596	6610	6613	6616	6619	6622
6628	6634	6637	6640	6643	6646	6649	6653	6656	6662	6669	6675	6681	6692	6725
6728	6732	6735	6738	6741	6744	6750	6756	6759	6762	6765	6768	6771	6774	6778
6784	6791	6797	6803	6817	6820	6823	6826	6829	6835	6841	6844	6847	6850	6852
6856	6859	6863	6869	6876	6882	6888	6899	6932	6935	6939	6942	6945	6948	6951
6957	6963	6966	6969	6972	6975	6978	6981	6985	6991	6998	7004	7010	7024	7027
7030	7033	7036	7042	7048	7051	7054	7057	7060	7063	7066	7070	7076	7083	7089
7095	7106	7139	7142	7146	7149	7152	7155	7158	7164	7170	7173	7176	7179	7182
7185	7188	7192	7198	7205	7211	7217	7231	7234	7237	7240	7243	7249	7255	7258
7261	7264	7267	7270	7273	7277	7283	7290	7296	7302	7313	7344	7347	7350	7353
7356	7359	7362	7365	7368	7374	7377	7380	7383	7386	7389	7392	7395	7398	7401
7404	7407	7413	7419	7425	7428	7431	7434	7437	7441	7450	7456	7462	7465	7468
7471	7475	7486	7518	7521	7524	7527	7530	7533	7536	7539	7542	7545	7548	7551
7554	7557	7560	7566	7569	7572	7575	7578	7581	7584	7587	7590	7593	7596	7599
7602	7605	7608	7611	7614	7623	7626	7629	7632	7635	7638	7641	7647	7650	7656
7662	7665	7668	7671	7674	7678	7687	7690	7696	7702	7705	7708	7711	7714	7718
7728	7761	7764	7767	7770	7773	7776	7779	7782	7785	7788	7791	7794	7797	7800
7803	7806	7809	7812	7815	7818	7821	7824	7839	7842	7846	7855	7858	7861	7865
7898	7901	7904	7907	7910	7913	7916	7919	7922	7925	7928	7931	7934	7937	7940
7943	7946	7961	7964	7967	7970	7973	7976	7979	7982	7985	7991	7994	8015	8018
8021	8024	8027	8030	8033	8036	8039	8042	8045	8048	8054	8060	8063	8066	8069
8072	8076	8104	8139	8142	8145	8148	8151	8154	8157	8160	8164	8167	8170	8176
8182	8185	8188	8191	8194	8197	8201	8207	8214	8220	8226	8241	8247	8254	8257
8264	8296	8299	8302	8305	8308	8311	8314	8317	8321	8324	8327	8333	8339	8342
8345	8348	8351	8354	8358	8364	8371	8377	8383	8401	8407	8414	8417	8424	8456
8459	8462	8465	8468	8471	8474	8477	8483	8489	8492	8495	8498	8501	8504	8507
8511	8521	8536	8542	8552	8555	8558	8561	8564	8567	8573	8579	8582	8585	8588
8591	8595	8605	8620	8636	8637	8649	8672	8675	8678	8691	8694	8687	8690	8702
8712	8715	8718	8721	8724	8727	8733	8736	8742	8745	8748	8751	8755	8761	8768
8774	8786	8792	8803	8835	8838	8841	8844	8847	8850	8853	8856	8859	8868	8875
8878	8882	8892	8904	8907	8919	8926	8961	8964	8967	8970	8973	8976	8979	8985
8988	8991	8994	8997	9000	9003	9006	9009	9012	9015	9018	9021	9024	9027	9030
9033	9036	9039	9042	9045	9048	9051	9054	9057	9060	9063	9066	9069	9072	9075
9078	9081	9084	9088	9092	9098	9101	9107	9113	9116	9119	9122	9125	9128	9132
9138	9145	9151	9157	9160	9163	9169	9172	9178	9184	9187	9190	9193	9196	9199
9203	9209	9216	9222	9228	9243	9278	9281	9284	9287	9290	9293	9296	9302	9305
9308	9311	9314	9317	9320	9323	9326	9329	9332	9335	9338	9341	9344	9347	9350
9353	9356	9359	9362	9365	9368	9371	9374	9377	9380	9383	9386	9389	9392	9395
9398	9401	9405	9409	9415	9418	9424	9430	9433	9436	9439	9442	9445	9449	9455
9462	9468	9474	9477	9480	9486	9489	9495	9501	9504	9507	9510	9513	9516	9520
9526	9533	9539	9545	9560	9595	9598	9601	9604	9607	9610	9613	9619	9622	9625
9628	9631	9634	9637	9640	9643	9646	9649	9652	9655	9658	9661	9664	9667	9670
9673	9676	9679	9682	9685	9688	9691	9694	9697	9700	9703	9706	9709	9712	9715
9718	9722	9726	9732	9735	9741	9747	9750	9753	9756	9759	9762	9766	9772	9779
9785	9791	9794	9797	9803	9806	9812	9818	9821	9824	9827	9830	9833	9837	9843
9850	9856	9862	9877	9912	9915	9918	9921	9924	9927	9930	9936	9939	9942	9945

9948	9951	9954	9957	9960	9963	9966	9969	9972	9975	9978	9981	9984	9987	9990
9993	9996	9999	10002	10005	10008	10011	10014	10017	10020	10023	10026	10029	10032	10035
10039	10043	10049	10052	10058	10064	10067	10070	10073	10076	10079	10083	10089	10096	10102
10108	10111	10114	10120	10123	10129	10135	10138	10141	10144	10147	10150	10154	10160	10167
10173	10179	10184	10239	10242	10245	10248	10241	10244	10247	10253	10256	10259	10262	10265
10268	10271	10274	10277	10280	10283	10286	10289	10292	10295	10298	10301	10304	10307	10310
10313	10316	10319	10322	10325	10328	10331	10334	10337	10340	10343	10346	10349	10352	10356
10360	10366	10369	10375	10381	10384	10387	10390	10393	10396	10400	10406	10413	10419	10425
10428	10431	10437	10440	10446	10452	10455	10458	10461	10464	10467	10471	10477	10484	10490
10496	10511	10546	10549	10555	10558	10558	10561	10564	10570	10573	10576	10579	10582	10585
10588	10591	10594	10597	10600	10603	10606	10609	10612	10615	10618	10621	10624	10627	10630
10633	10636	10639	10642	10645	10648	10651	10654	10657	10660	10663	10666	10669	10673	10677
10683	10686	10692	10698	10701	10704	10707	10710	10713	10717	10723	10730	10736	10742	10745
10748	10754	10757	10763	10769	10772	10775	10778	10781	10784	10788	10794	10801	10807	10813
10828	10863	10866	10869	10872	10875	10878	10881	10887	10890	10893	10896	10899	10902	10905
10908	10911	10914	10917	10920	10923	10926	10929	10932	10935	10938	10941	10944	10947	10950
10953	10956	10959	10962	10965	10968	10971	10974	10977	10980	10983	10986	10990	10994	11000
11003	11009	11015	11018	11021	11024	11027	11030	11034	11040	11047	11053	11059	11062	11065
11071	11074	11080	11086	11089	11092	11095	11098	11101	11105	11111	11118	11124	11130	11145
11180	11183	11186	11189	11192	11195	11198	11204	11207	11210	11213	11216	11219	11222	11225
11228	11231	11234	11237	11240	11243	11246	11249	11252	11255	11258	11261	11264	11267	11270
11273	11276	11279	11282	11285	11288	11291	11294	11297	11300	11303	11307	11311	11317	11320
11326	11332	11336	11338	11341	11344	11347	11361	11357	11364	11370	11376	11379	11382	11388
11391	11397	11403	11406	11409	11412	11415	11418	11422	11428	11435	11441	11447	11462	11497
11500	11503	11506	11509	11512	11515	11521	11524	11527	11530	11533	11536	11539	11542	11545
11548	11551	11554	11557	11560	11563	11566	11569	11572	11575	11578	11581	11584	11587	11590
11593	11596	11599	11602	11605	11608	11611	11614	11617	11620	11624	11628	11634	11637	11643
11649	11652	11655	11658	11661	11664	11668	11624	11681	11687	11693	11696	11699	11705	11708
11714	11720	11723	11726	11729	11732	11735	11739	11745	11752	11758	11764	11779	11814	11817
11820	11823	11826	11829	11832	11835	11841	11844	11847	11850	11853	11856	11859	11862	11865
11868	11871	11874	11877	11880	11883	11886	11889	11892	11895	11898	11901	11904	11907	11910
11913	11916	11919	11922	11925	11928	11931	11934	11937	11941	11945	11951	11954	11960	11966
11969	11972	11975	11978	11981	11984	11991	11998	12004	12010	12013	12016	12022	12025	12031
12037	12040	12043	12046	12049	12052	12056	12062	12069	12075	12081	12096	12131	12134	12137
12140	12143	12146	12149	12155	12158	12161	12164	12167	12170	12173	12176	12179	12182	12185
12188	12191	12194	12197	12200	12203	12206	12209	12212	12215	12218	12221	12224	12227	12230
12233	12236	12239	12242	12245	12248	12251	12254	12258	12262	12268	12271	12277	12283	12286
12289	12292	12295	12298	12302	12305	12315	12321	12324	12330	12333	12339	12342	12348	12354
12357	12360	12363	12366	12369	12373	12379	12386	12389	12392	12413	12418	12451	12454	12457
12460	12463	12466	12472	12475	12478	12481	12484	12487	12490	12493	12496	12499	12502	12505
12508	12511	12514	12517	12520	12523	12526	12529	12532	12536	12538	12541	12544	12547	12550
12553	12556	12559	12562	12565	12568	12571	12575	12578	12582	12585	12591	12600	12603	12606
12609	12612	12615	12619	12622	12626	12638	12641	12647	12650	12656	12659	12665	12671	12674
12677	12680	12683	12686	12689	12692	12703	12709	12715	12730	12765	12768	12771	12774	12777
12780	12783	12789	12792	12795	12798	12801	12804	12807	12810	12813	12816	12819	12822	12825
12828	12831	12834	12837	12840	12843	12846	12849	12852	12855	12858	12861	12864	12867	12870
12873	12876	12879	12882	12885	12888	12892	12896	12902	12905	12911	12917	12920	12923	12926
12929	12932	12936	12943	12946	12955	12961	12964	12967	12973	12976	12982	12988	12991	12994
12997	13000	13003	13007	13013	13020	13026	13032	13047	13082	13085	13088	13091	13094	13097
13100	13106	13109	13112	13115	13118	13121	13124	13127	13130	13133	13136	13139	13142	13145
13148	13151	13154	13157	13160	13163	13166	13169	13172	13175	13178	13181	13184	13187	13190
13193	13196	13199	13202	13205	13209	13213	13219	13222	13228	13234	13237	13240	13243	13246
13249	13253	13254	13266	13272	13278	13281	13284	13290	13293	13296	13305	13308	13311	13314
13317	13320	13324	13330	13337	13343	13349	13364	13399	13402	13405	13408	13411	13414	13417
13423	13426	13429	13432	13435	13438	13441	13444	13447	13450	13453	13456	13459	13462	13465
13468	13471	13474	13477	13480	13483	13486	13489	13492	13495	13498	13501	13504	13507	13510

```
            13513  13516  13519  13522  13526  13530  13536  13539  13545  13551  13554  13557  13560  13563  13566
            13570  13576  13583  13589  13595  13602  13601  13607  13610  13616  13622  13625  13628  13631  13634
            13637  13641  13647  13654  13660  13666  13681  13714  13717  13720  13723  13726  13729  13735  13739
            13742  13751  13754  13757  13767  13770  13774  13783  13795  13805  13808  13812  13822  13857  13860
            13863  13866  13870  13873  13876  13879  13882  13885  13888  13892  13895  13900  13903  13906  13909
            13912  13915  13918  13921  13927  13930  13933  13936  13939  13942  13945  13948  13951  13954  13957
            13962  13971  13974  13980  13990  13996  14009  14021  14027  14031  14034  14046  14049  14053  14059
            14063  14069  14073  14079  14083  14089  14093  14099  14103  14109  14113  14121  14125  14131  14135
            14141  14145  14151  14155  14161  14165  14171  14175  14181  14185  14191  14195  14204  14207  14211
            14217  14220  14223  14226  14230  14236  14239  14243  14249  14252  14256  14262  14265  14269  14275
            14278  14282  14288  14291  14295  14301  14304  14308  14314  14317  14321  14327  14330  14334  14343
            14346  14349  14352  14355  14361  14367  14370  14373  14376  14380  14408  14417  14420  14426  14432
            14435  14438  14441  14444  14448  14481  14484  14487  14490  14493  14496  14499  14502  14505  14508
            14511  14514  14517  14520  14523  14526  14529  14532  14535  14538  14541  14544  14550  14553  14556
            14562  14565  14568  14574  14580  14583  14589  14595  14598  14604  14607  14613  14623  14626  14632
            14638  14641  14644  14647  14650  14654  14664  14699  14702  14705  14708  14711  14714  14717  14720
            14723  14726  14729  14732  14738  14747  14750  14753  14758  14784  14815  14818  14821  14824  14827
            14830  14833  14836  14839  14842  14845  14851  14860  14863  14866  14869  14872  14891  14894  14897
            14900  14903  14906  14909  14912  14915  14921  14930  14933  14936  14939  14942  14955  14958  14961
            14964  14967  14970  14973  14976  14979  14985  14988  14991  14994  15010  15013  15016  15019  15022
            15025  15028  15034  15041  15047
.SETUP        18
.TRI          18  2306  2386  2503  2583  2700  2780  2897  2977  3092  3104  3181  3193  3309  3324
            3330  3407  3422  3438  3545  3625  3742  3822  3939  4019  4136  4216  4333  4413  4530
            4610  4727  4807  4924  5004  5121  5191  5301  5386  5508  5593  5715  5800  5922  6007
            6129  6214  6336  6421  6543  6628  6750  6835  6957  7042  7164  7249  7419  7456  7554
            7611  7656  7696  7815  7861  8054  8176  8247  8333  8407  8483  8542  8573  8626  8702
            8733  8792  9056  9107  9166  9178  9231  9412  9418  9424  9483  9486  9495  9548  9729
            9735  9741  9800  9803  9812  9865  9871  9877  9883  9889  9895  9901  9907  9913  9919
          10375  10434  10437  10446  10499  10680  10686  10692  10751  10754  10763  10816  10997  11019  11068
          11080  11133  11314  11320  11326  11385  11388  11397  11450  11631  11637  11643  11702  11705  11714
          11762  11948  11954  11960  12019  12025  12031  12084  12265  12277  12336  12348  12401  12582  12594
          12653  12665  12718  12899  12911  12970  12982  13035  13216  13222  13228  13287  13290  13299  13352
          13533  13545  13604  13616  13669  13680  14361  14536  14539  14541  14613  14632  14720  14729  14753
          14827  14836  14842  14863  14906  14912  14933  14970  14976  14991
.UNI          18  5159  5162  5229  5232  7830  7952  7997  8003  8006  8009  8065  8071  8094  8097
            8235  8238  8395  8398  8530  8533  8614  8617  8693  8696  8780  8783  8862  8865  8901
            8910  8913  8916  8982  9233  9295  9550  9616  9867  9933  10184  10250  10501  10567  10818
          10884  11003  11071  11135  11201  11452  11518  11769  11835  12086  12152  12271  12339  12403  12469
          12588  12656  12720  12786  12905  12973  13037  13103  13354  13420  13539  13607  13671  13789  13792
          13965  13968  14003  14012  14037  14389  14398  14767  14770  14773  14809  14812  14885  14888
.SACT1        18  217
.SAPTB        18  295#
.SAPTH        18  481
.SAPTY        18  1174
.SCATC        18
.SCNTA        18  240
.SEOP         18  959
.SERRO        18
.SERRT        18
.SPOME        18  1662
.SRDOC        18  1336
.SREAD        18  1233
.SSCOP        18  1026
.STRAP        18  1521
.STYPE        18  1095
```

```
.STYPO      1#


. ABS.  037470     000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

DZKCA,DZKCA/SOL/CRF+DZKCA.MAK,DZKCA.MAC,DZKCA.P11/EQ:DZKCA
RUN-TIME: 219 224 13 SECONDS
RUN-TIME RATIO: 7717/457=16.8
CORE USED:  55K  (110 PAGES)
```